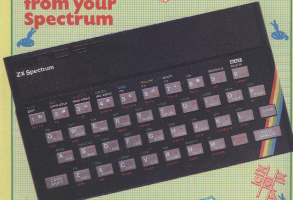
Spring 1984

An Argus Specialist Publication

Personal SOFTWARE

£2.10

Get more from your Spectrum



- Great games to test your reflexes
- Learn morse code
- Go for different graphics
- Play cards, win wars and destroy aliens
- Useful utilities to run

ANIROG SOFTWARE

COMPUTER GAMES OF TOMORROW AVAILABLE NOW!

AS SEEN ON T.V.!

HEXPERT

MOON BUGGY

SKRAMBLE

3D TIME TREK

COMMODORE 64

NEW

GALAXY

AVOID CAPTURE BY THE ALIEN MOTHER SHIPS TRACTOR BEAM AS THE FIGHTERS DIVE BOMB YOU. 100 SCREENS WITH A TWO PLAYER OPTION. K.B./J.S. £7.95

KONG K.B./J.S. £7.95
4 SCREENS WITH BRILLIANT ANIMATED GRAPHICS
SKRAMBLE J.S. £7.95
6 SECTORS TO TEST YOUR SKILL
HEXPERT J.S. £7.95
TRAIN BERT TO BE HEXPERT ON THIS 3D HEXAGONAL PYRAMID
MOON BUGGY J.S. £7.95
MANOEUVRE YOUR PATROL CRAFT OVER GIANT POT HOLES AS YOU DEFEND THE MOON BUGGY FROM ALIEN ATTACK
FROG RUN K.B./J.S. £5.95
A POPULAR ARCADE GAME. FUN FOR THE WHOLE FAMILY
3D TIME TREK K.B./J.S. £5.95
SPECTACULAR 3D GRAPHICS STAR TREK GAME
DUNGEONS K.B. £6.95
ENTER THE REALMS OF FANTASY IN THIS ROLE PLAYING GAME
DARK DUNGEONS K.B. £6.95
2ND IN THE SERIES OF FOUR. DEFINITELY NOT FOR THE FAINT HEARTED.

NEW APPROVED GAMES FOR THE STACK LIGHT RIFLE

INDIAN ATTACK £5.95
COSMIC COMMANDO £5.95
VENGEANCE OF ZENO £5.95

24 HR. CREDIT CARD SALES HORLEY (02934) 6083
PAYMENT BY CHEQUE, P.O., ACCESS/VISA
8 HIGH STREET HORLEY, SURREY.
Overseas 50p post & packaging

TRADE ENQUIRIES WELCOME
29, West Hill, Dartford, Kent. (0322) 92513/8

Volume 2
No 4
Spring 1984

Editor: Wendy J Palmer
Sales Executive:
David Poulter
Advertisement Manager:
Barry Bingham
Advertisement Copy Control:
Ann McDermott
Managing Editor:
Ron Harris
Chief Executive: T J Connell
Origination and design by MM Design & Print

Personal Software is a quarterly magazine appearing on the third Friday in May, August, November and February.

Distribution by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4IS Tel: 01-247 8233
Printing by: Alabaster Passmore & Sons Ltd, Tovil, Maidstone, Kent.

The contents of this publication including all articles, designs, plans, drawings and programs and all copyright and other intellectual property rights therein belong to Argus Specialist Publications Limited. All rights conferred by the Law of Copyright and other intellectual property rights and by virtue of international copyright conventions are specifically reserved to Argus Specialist Publications Limited and any reproduction requires the prior written consent of the Company. © 1984 Argus Specialist Publications Limited.

Subscription Rates : UK £8.40 for four issues including postage. Airmail and other rates upon application to Personal Software Subscriptions, PO Box 35, Wolsey House, Wolsey Road, Hemel Hempstead, Herts HP2 4SS.

CONTENTS

Editorial & Advertisement Office
No 1 Golden Square, London W1R 3AB
Telephone: 01-437 0626 Telex: 8811896

Preface 4

Battleships 5
Command the fleet and sink the enemy's ships.

Tourist Trap 10
Ever had one of those disastrous holidays frought with problems? See how you would cope.

Planet 19
An educational game for the Spectrum.

Play Your Cards Right 22
See if you can deal yourself a good hand.

Demolition 26
Get rid of your destructive tendencies without having to pay for the damage.

Drawing on Experience 28
A tutorial on the subject of on-screen drawing including a bar chart program and a rather sophisticated sketching listing.

Earth Attack 31
Can you beat off the attack and save the earth?

Toolbox Trio 36
Three useful routines for your Spectrum.

Spectrum Breakout 38
Demolish the wall and be as aggressive as you like.

Alternative Characters 40
Fed up with the characters on the Spectrum? Re-define them with this utility.

Statistical Analysis 41
Get to grips with numbers and statistics.

Morse Trainer 48
Talk to your Spectrum in morse code.

Aliens 50
Stop the alien force or you could find yourself in trouble.

Leprechaun's Gold 52
Are you lucky enough to find the gold?

Circuit Sketch 57
Great definition when drawing circuit diagrams.

Bats 60
A batty little game to get your Spectrum going.

3D Noughts and Crosses 63
Throw away your pencil and paper and play on your Spectrum.

Air Raiders 68
This game will have you shooting down the enemy planes and missiles.

Drainpipes 72
This game may quickly drain your energies.

Machine Code Colour Graphics 74
Machine Code isn't really difficult — nor are colour graphics.

Getting Fizzical 78
Make money by selling lemonade.

POKEing the Spectrum Display 80
What a fine display this utility makes.

Asteroids Ahead 82
Steer clear of the asteroids and keep your spaceship flying.

Learner Driver 85
Will you pass the test?

Defending Your Spectrum 88
Let this game knock down your defences.

Spectramon 90
An incredible disassembler for the 48K Spectrum.

Credits: Our grateful thanks to the following individuals for their assistance in the production of this issue. Mark, Gary, Bill, Bill, Nigel, Tom, William, Adrian, Jane, Fiona and others almost too numerous to mention.

Note: Sinclair, ZX, ZX Spectrum, ZX Microdrive, ZX Interface are all registered trademarks of Sinclair Research Ltd.

PREFACE

All right then, hands up everyone who received a Spectrum from Santa Claus this Christmas. And hands up all of you who have beaten the games that you bought to play on your new 'toy'.

It is true to say that with micros, once used, forever addicted. And you can't wait to get home with the new game you've just bought to see if you can defend the earth, destroy the aliens and sink the enemy's battleships — and all before tea! The Spectrum is so popular with the home micro user that there is no lack of commercial software available for you to test your wits and reflexes. Sadly, though, few of us have a bottomless pocket and the standard of software around is not universally high.

It is also probably true to say that although a large number of Spectrum owners start their computing careers wanting to play games, they soon learn that there is more to micros than just splatting aliens. They want to find out more about how their Spectrum works, and they start tentatively writing their own programs (games included).

In this issue of ***Personal Software*** we have drawn together some of the best programs for the Spectrum that have been published in our sister magazine, ***ZX Computing***.

The following pages are packed with games for you to type in — from the more 'old-fashioned' game of noughts and crosses to the current 'trend' of Aliens and Earth Attack. For those who are not particularly interested in games, there are some useful utilities and programs such as Spectramon and Alternative Characters. Certainly there should be something for everyone in this issue!

CHECKING UP

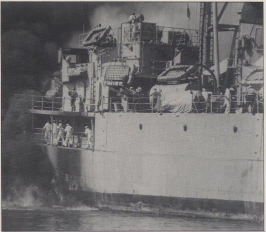
We have tried to ensure that the programs in this issue are error-free, and that they are legible, so if you find that you are having problems with any of them, please check them again thoroughly, or get someone else to check them for you. If as a last resort you can still not get the program running, then write to ***Personal Software*** with a SAE. Please note however that we are unable to answer telephone enquiries.

ZX Spectrum

Jeff Hamilton

BATTLESHIPS

Command your own fleet and sink the enemy's ships.



This program has been written to allow users to challenge their Spectrum computer to the age-old classroom game of Battleships.

When you have the game RUNning, you will be asked to INPUT the direction of your ships and the starting square. The Spectrum will automatically set your ships on the grid. Once, a battleship (four squares), two cruisers (three squares) and three destroyers (two squares) have been set up, the Spectrum decides who should begin the game.

The Spectrum's moves are calculated in lines 1000 to 1080; a random square is chosen to fire at unless the Spectrum has just scored a hit on your fleet. If it has just achieved damage to one of your ships, it will continue to fire at the surrounding squares until it has sunk the vessel.

IN THE NAVY

Once the Spectrum has selected a square to fire at, it is up to the user to respond with either a 'M' for a miss or an 'H' for a direct hit. There is no routine in the program to counteract the user lying to the computer, but there is no reason why a simple routine could not be written to stop any 'innocent' cheating.

When it is the user's turn, simply INPUT the co-ordinates, for example, 'F7' for the grid reference F7, and the result will be displayed on the screen as a miss or as a hit. It might be an idea to make a mental note as to which squares you have already fired at, as the computer will not tell you if you have already had a go at a certain grid reference.

▶

```
  50 GO SUB 6000: REM ** RULES *
*
 100 GO SUB 5000: REM ** INITIAL
ISE VARIABLES **
 110 CLS : GO SUB 5300: REM ** S
ET UP BOARD **
 120 GO SUB 6500: REM ** SET UP
3RD SQUARE **
 130 GO SUB 5400: REM ** HIDE SH
IPS **
 135 LET Z=INT (RND*2): IF Z=1 T
HEN GO TO 150
 140 GO TO 700: REM ** PLAYER GO
**
 150 GO TO 1000: REM ** SPECTRUM
 GO **
 700 PRINT AT 14,14; PAPER 1; IN
K 7;"YOUR  GO"
 710 INPUT D$: IF LEN D$<>2 THEN
 GO TO 710
 715 LET B$=D$(1): LET C=VAL D$(
2)
 720 LET D=CODE B$-64
 725 IF (C<0 OR C>9) OR (D<1 OR
D>10) THEN GO TO 710
 730 LET W=D: LET N=C+1
 740 LET J=M(N,W)
 750 IF M(N,W)<>143 THEN LET HIT
S=HITS+1
 770 PRINT INK 2;AT C,D;CHR$ J:
IF J>100 THEN LET J=J-100
 775 BEEP .05,J-40
 780 IF HITS=16 THEN GO TO 4000
 790 GO TO 150
1000 IF LB=0 AND LC=0 AND LD=0 T
HEN GO SUB 1500
1010 IF LB=1 OR LC=1 OR LD=1 THE
N GO SUB 1800
1020 IF LB>1 OR LC>1 THEN GO SUB
 2000
1030 INPUT M$: IF NOT M$="H" AND
 NOT M$="M" THEN GO TO 1030
1040 IF M$="M" THEN GO SUB 1550
1050 IF M$="H" THEN GO SUB 1580
1060 PRINT INK 2;AT X-4,Y+9;CHR$
 J: IF J>100 THEN LET JJ=J-100
1061 IF J<100 THEN LET JJ=J
1062 PRINT ; INVERSE 1; INK 1; F
LASH 0;AT X+7,Y-3;CHR$ J
1063 IF J>100 THEN PRINT ; INVER
SE 0; INK 1; FLASH 0;AT X+7,Y-3;
CHR$ J
1064 BEEP .05,JJ-40
1065 LET L(X,Y)=J
1070 IF HATS=16 THEN GO TO 4020
1080 GO TO 140
1500 LET X=INT (RND*10)+4: LET Y
=INT (RND*10)+4
1510 IF L(X,Y)<>46 THEN GO TO 15
00
1515 IF L(X+1,Y)<>46 AND L(X,Y-1
)<>46 AND L(X-1,Y)<>46 AND L(X,Y
+1)<>46 THEN GO TO 1500
1520 LET YA=Y+61
1530 PRINT AT 14,14; PAPER 2; IN
K 7;"MY GO ";CHR$ YA;X-4
1535 PRINT ; BRIGHT 1; INK 1; FL
ASH 1;AT X+7,Y-3;"?"
1540 RETURN
1550 LET J=143
1560 IF LB=1 OR LC=1 OR LD=1 THE
N LET LZ=LZ+1
1570 RETURN
1580 PRINT AT 16,14; PAPER 4; IN
K 7;"B,C OR D?"
1590 INPUT N$: FOR N=1 TO 9: PRI
NT AT 16,N+13; PAPER 7; INK 7;CH
R$ 32;: NEXT N
1600 LET HATS=HATS+1
1610 IF N$="B" THEN GO SUB 1750
1620 IF N$="C" THEN GO SUB 1700
1630 IF N$="D" THEN GO SUB 1650
1640 RETURN
1650 LET J=CODE "D"
1660 LET LD=LD+1
1670 IF LD=2 THEN LET LZ=0
1680 IF LD=2 THEN LET LD=0
1690 RETURN
1700 LET J=CODE "C"
1710 LET LC=LC+1
1720 IF LC=3 THEN LET LZ=0: LET
LY=0
1730 IF LC=3 THEN LET LC=0
1740 RETURN
1750 LET J=CODE "B"
1760 LET LB=LB+1
1770 IF LB=4 THEN LET LZ=0: LET
LY=0: LET LR=0
1780 IF LB=4 THEN LET LB=0
1790 RETURN
1800 IF (LB=1 OR LC=1) AND LZ=0
THEN IF (L(X-2,Y)<>46 AND L(X+1,
Y)<>46) OR (L(X-1,Y)<>46 AND L(X
+2,Y)<>46) THEN LET Y=Y-1: LET L
Z=2: GO TO 1970
1810 IF (LB=1 OR LC=1) AND LZ=1
THEN IF L(X+3,Y)<>46 THEN LET X=
X+1: LET Y=Y-1: LET LZ=2: GO TO
1970
1850 IF LZ=0 THEN GO TO 1890
1860 IF LZ=1 THEN GO TO 1930
1870 IF LZ=2 THEN GO TO 1965
1880 IF LZ=3 THEN GO TO 1985
1890 LET X=X-1
1900 IF L(X,Y)<>46 THEN GO TO 19
20
1910 LET LZD=1: GO TO 1990
1920 LET LZ=LZ+1
1930 LET X=X+2
1940 IF L(X,Y)<>46 THEN GO TO 19
60
1950 LET LZD=2: GO TO 1990
1960 LET LZ=LZ+1
1965 LET X=X-1: LET Y=Y-1
1970 IF L(X,Y)<>46 THEN GO TO 19
80
1975 LET LZD=3: GO TO 1990
1980 LET LZ=LZ+1
1985 LET Y=Y+2: LET LZD=4
1990 GO SUB 1520
1995 RETURN
2000 IF LZD=1 THEN GO SUB 2050
2010 IF LZD=2 THEN LET X=X+1
2020 IF LZD=3 THEN GO SUB 2080
2030 IF LZD=4 THEN LET Y=Y+1
2040 GO SUB 1520: RETURN
2050 IF LB=2 OR LC=2 THEN GO SUB
 2110
2060 IF LB=3 THEN GO SUB 2300
2070 RETURN
2080 IF LB=2 OR LC=2 THEN GO SUB
 2160
2090 IF LB=3 THEN GO TO 2400
2100 RETURN
2110 IF (LB=2 OR LC=2) AND LY=1
THEN LET X=X+3: RETURN
2120 LET X=X-1
2130 IF X=3 THEN GO TO 2150
2140 IF L(X,Y)=46 THEN LET LY=LY
+1: RETURN
2150 IF LB=2 OR LC=2 THEN LET X=
X+3: RETURN
2160 IF (LB=2 OR LC=2) AND LY=1
THEN LET Y=Y+3: RETURN
2170 LET Y=Y-1
2180 IF Y=3 THEN GO TO 2200
2190 IF L(X,Y)=46 THEN LET LY=LY
+1: RETURN
2200 IF LB=2 OR LC=2 THEN LET Y=
Y+3: RETURN
2300 IF LR=1 THEN LET X=X+4: RET
URN
2310 LET X=X-1
2320 IF X=3 THEN LET X=X+4: RETU
RN
2330 IF L(X,Y)<>46 THEN LET X=X+
2: RETURN
2340 IF L(X,Y)=46 THEN LET LR=1:
 RETURN
2400 IF LR=1 THEN LET Y=Y+4: RET
URN
2410 LET Y=Y-1
2420 IF Y=3 THEN LET Y=Y+4: RETU
RN
2430 IF L(X,Y)<>46 THEN LET Y=Y+
2: RETURN
2440 IF L(X,Y)=46 THEN LET LR=1:
 RETURN
4000 PRINT AT 16,14;"YOU WON": G
O TO 4040
4020 PRINT AT 16,14;" I WON ": G
O TO 4040
4040 PRINT AT 18,14;"AGAIN? (Y/N
)"
4050 INPUT Y$
4055 IF NOT Y$="Y" THEN GO TO 40
```

```
70
4060 IF Y$="Y" THEN RUN
4070 PRINT AT 21,14;"OK BYE": GO
 TO 9000
5000 REM ** INITIALISE **
5010 POKE 23609,100
5020 DIM L(16,16)
5025 DIM M(10,10)
5027 FOR N=1 TO 10: FOR W=1 TO 1
0
5028 LET M(N,W)=143
5030 NEXT W: NEXT N
5035 RANDOMIZE
5040 LET HITS=0: LET HATS=0
5050 LET LB=0: LET LC=0: LET LD=
0: LET LZ=0: LET LA=0: LET LZD=0
5070 LET LX=0: LET LY=0: LET XX=
0: LET LR=0
5080 FOR X=1 TO 3: FOR Y=1 TO 3
5090 LET L(X,Y)=100: NEXT Y: NEX
T X
5100 FOR X=4 TO 13: FOR Y=4 TO 1
3
5110 LET L(X,Y)=46: NEXT Y: NEXT
 X
5120 FOR X=14 TO 16: FOR Y=14 TO
 16
5130 LET L(X,Y)=100: NEXT Y: NEX
T X: RETURN
5300 REM ** SET UP BOARD **
5310 FOR A=0 TO 9
5320 PRINT INK 1;A; PAPER 6; INK
 2;".........."; PAPER 7;"  "; P
APER 6;".........."; INK 1; PAPE
R 7;A
5330 NEXT A
5340 PRINT INK 1;" ABCDEFGHIJ  A
BCDEFGHIJ "
5350 FOR A=0 TO 9
5360 PRINT INK 1;A; PAPER 6; INK
 2;"..........": NEXT A
5370 PLOT 8,8: GO SUB 5395
5380 PLOT 8,96: GO SUB 5395
5390 PLOT 104,96: GO SUB 5395: R
ETURN
5395 DRAW INK 2;79,0: DRAW INK 2
;0,79: DRAW INK 2;-79,0: DRAW IN
K 2;0,-79: RETURN
5400 REM ** HIDE SHIPS **
5410 LET AB=6: GO SUB 5600: FOR
P=1 TO 4
5420 LET M(N,W)=CODE "B"
5430 GO SUB 5640: NEXT P
5440 FOR E=1 TO 2
5450 LET AB=7: GO SUB 5600: GO S
UB 5670: IF A=1 THEN GO TO 5450
5460 FOR P=1 TO 3
5470 LET M(N,W)=CODE "C"
5480 GO SUB 5640: NEXT P: NEXT E
5490 FOR E=1 TO 3
5500 LET AB=8: GO SUB 5600: GO S
UB 5700: IF A=1 THEN GO TO 5500
5510 FOR P=1 TO 2
5520 LET M(N,W)=CODE "D"
5530 GO SUB 5640: NEXT P: NEXT E
5540 RETURN
5600 LET A=0: LET AA=INT (RND*2)
5610 IF AA=0 THEN LET N=INT (RND
*AB)+1: LET W=INT (RND*10)+1
5620 IF AA=1 THEN LET N=INT (RND
*10)+1: LET W=INT (RND*AB)+1
5630 RETURN
5640 IF AA=0 THEN LET N=N+1
5650 IF AA=1 THEN LET W=W+1
5660 RETURN
5670 IF AA=0 THEN IF M(N,W)<>143
 OR M(N+1,W)<>143 OR M(N+2,W)<>1
43 THEN LET A=1
5680 IF AA=1 THEN IF M(N,W)<>143
 OR M(N,W+1)<>143 OR M(N,W+2)<>1
43 THEN LET A=1
5700 IF AA=0 THEN IF M(N,W)<>143
 OR M(N+1,W)<>143 THEN LET A=1
5710 IF AA=1 THEN IF M(N,W)<>143
 OR M(N,W+1)<>143 THEN LET A=1
5720 RETURN
6000 FOR Y=1 TO 7: BORDER Y: FOR
 N=1 TO 4: BEEP .005,N+Y
6100 PRINT ; PAPER N; INK N+2;AT
 2,5;"███ ███ ███ ███ █  ███"
6110 PRINT ; PAPER N; INK N+2;AT
 3,5;"█ █ █ █  █   █   █   █  "
6120 PRINT ; PAPER N; INK N+2;AT
 4,5;"███ ███  █   █   █   ███"
6130 PRINT ; PAPER N; INK N+2;AT
 5,5;"█ █ █ █  █   █   █   █  "
6140 PRINT ; PAPER N; INK N+2;AT
 6,5;"███ █ █  █   █   ███ ███"
6150 PRINT ; PAPER N; INK N+2;AT
 8,8;"███ █ █ █ ███ ███"
6160 PRINT ; PAPER N; INK N+2;AT
 9,8;"█   █ █ █ █ █ █  "
6170 PRINT ; PAPER N; INK N+2;AT
 10,8;"███ ███ █ ███ ███"
6180 PRINT ; PAPER N; INK N+2;AT
 11,8;"  █ █ █ █ █   █"
6190 PRINT ; PAPER N; INK N+2;AT
 12,8;"███ █ █ █ █   ███"
6195 PRINT ; PAPER N+1; INK N+3;
AT 14,4;"© J.H.W.HAMILTON 2/10/8
2"
6200 NEXT N: NEXT Y
6210 PAUSE 60
6220 PRINT ; PAPER 1; INK 7; FLA
SH 1;AT 16,0;"ALWAYS ENTER LETTE
R AND NUMBER        THEN  PRESS
ENTER           "
6230 PAUSE 60: GO SUB 6400
6240 PAUSE 4E4: GO SUB 6410
6250 PRINT ; PAPER 3; INK 6; FLA
SH 0;AT 16,0;"  FIRST  SET  UP
3RD  SQUARE      BY INPUTTING ONE
 BATTLESHIP   (4 SQUARES) TWO CR
UISERS (3 SQS) AND THREE DESTROY
ERS (2 SQS)     "
6260 PAUSE 60: GO SUB 6400
6270 PAUSE 4E4: GO SUB 6410
6280 PRINT ; PAPER 1; INK 7;AT 1
6,0;" WHEN IT IS YOUR GO (TOP LE
FT)          INPUT YOUR GUESS
     ON THE SPECTRUMS GO (TOP R
IGHT) INPUT-M-IF MISS;-H-IF HIT:I
F HIT  YOU WILL BE ASKED WHICH S
HIP  "
6290 PAUSE 60: GO SUB 6400
6300 PAUSE 4E4: GO SUB 6410: RET
URN
6400 PRINT ; FLASH 1;AT 21,8;"PR
ESS ENTER": RETURN
6410 PRINT FLASH 0;AT 21,8;"
      ": RETURN
6500 REM ** SET UP 3RD SQUARE **
6510 LET X$="BATTLESHIP (4)"
6520 LET DF=6: LET J=66
6530 LET X=4: GO SUB 6700
6540 LET X$="CRUISER (3)"
6550 LET DF=7: LET J=67: LET X=3
6560 GO SUB 6700: GO SUB 6700
6570 LET X$="DESTROYER (2)"
6580 LET DF=8: LET J=68: LET X=2
6590 GO SUB 6700: GO SUB 6700: G
O SUB 6700: RETURN
6600 PRINT AT 14,17;"     ";AT 1
5,15;"          ";AT 16,15;"
      ";AT 17,18;"   ";AT 18,13;"
 
6610 RETURN
6630 IF BR=1 THEN LET C=C+1
6640 IF BR=2 THEN LET D=D+1
6650 RETURN
6700 PRINT AT 14,17; INK 3;"INPU
T";AT 15,15;"1 FOR N/S";AT 16,15
;"2 FOR E/W"
6710 INPUT BR: IF BR<1 OR BR>2 T
HEN GO TO 6710
6720 IF BR=1 THEN LET DD=DF: LET
 DE=9
6730 IF BR=2 THEN LET DD=9: LET
DE=DF
6740 PRINT AT 14,17; INK 4;"ENTE
R";AT 15,15;"STARTING ";AT 16,15
;" SQUARE  ";AT 17,18;"FOR";AT 1
8,13;X$
6750 INPUT A$: IF LEN A$<>2 THEN
 GO TO 6750
6760 LET B$=A$(1): LET C=VAL A$(
2)
6770 LET D=CODE B$-65
6780 IF C<0 OR C>DD THEN GO TO 6
750
6790 IF D<0 OR D>DE THEN GO TO 6
750
6800 GO SUB 6600
6810 FOR F=1 TO X
6820 PRINT AT C+11,D+1; INK 2;CH
R$ J
6830 GO SUB 6630: NEXT F: RETURN
```

+ + +

OMNICALC (48K Spectrum)
The best Spectrum spreadsheet you can buy. Written in machine code to be faster, to give you more space for data, and to allow more features to be included, it is guaranteed uncrashable.

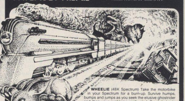
- ★ 99 columns or 250 rows.
- ★ Fully prompted input.
- ★ Extensive repeat facilities.
- ★ Supports ALL Spectrum functions.
- ★ Unique iterative facility.
- ★ Totals/Sub-totals.
  + many more.

"If more programs shared the quality of OMNICALC then we might see more Spectrums in offices"
Home Computing Weekly
3/6/83

"It is ideal for someone who has just found the spreadsheet concept, but it is also a very powerful tool for anyone who has used one previously"
Sinclair User June 1983

Complete with comprehensive manual. **£9.95.**

MICROSPHERE 72 ROSEBERY ROAD · LONDON N10 2LA
TELEPHONE 01-883 9411

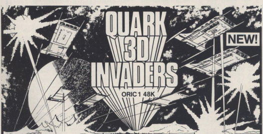


WHEELIE (48K Spectrum) Take the motorbike in your Spectrum for a burn-up. Survive humps, bumps and jumps as you seek the elusive ghostrider — and then your problems really start. Crash helmets advisable. **£5.95**

EVOLUTION (48K Spectrum) Tyrannosaurus Rex, Stegosaurus, Podopteryx, Brontosaurus are some of the creatures on the fascinating journey from the start of life to man. Educational and fun, this adventure took 3500 million years the first time it was played . . . Can you do it quicker? . . . — just **£6.95**

CREVASSE/HOTFOOT (Any Spectrum) Can you make it across the ice-cap whilst chasms open and close all around you . . . or help a hungry rabbit in a hostile world . . . 2 original games on one cassette — only **£4.95**

THE TRAIN GAME (Any Spectrum) The game that starts where model railways left off. Full graphics featuring passenger, goods & express trains; tunnels; stations; turntable bonus games; irate passengers; collisions; derailments; and everything else you expect from a major railway! — Superb value **£5.95**



Super 3D arcade action Invaders will test your skill and 3D co-ordination to their limits. 8 rippling colour invader screens of increasing difficulty with keyboard controlled 3D laser movement and fire sequence. (Brain required) **£6.95**

DRAGON 32
MRC £6.50 • Fruit £4.95 • Look and Learn £4.95 • Pontoon £4.95
Missile Command £4.95 • Graphics Demonstrator £4.95
Tango Foxtrot 01 £6.50 • Space Crystal £6.95

ORIC 1 48K
Oricle (Smart Oric) £6.95 • Flight Simulator £6.95 • Super Fruit £6.95 • Space Crystal £6.95

QUARK DATA P.O. BOX 61, Swindon, Wilts. Tel. (0793) 40661.
Please supply . . . .
. . . .
Prices incl. p & p UK only.
Cheque for total amount enclosed . . . .
Name . . . .
Address . . . .
. . . .

Mr. CHIP SOFTWARE

SPECTRUM GAMES

SPECTRUM DARTS (48K)

Five games of darts for 1-5 players, 501, Cricket, Killer, Round the board, Noughts and Crosses, four levels of play per game, take on the computer or friends at these games of skill and judgement **£5.50**

WH££LER DEALER

As for the Commodore 64, but now available for the 48K Spectrum, Texas T199/4A and Dragon **£5.50**

VIC-20
GAMES AND UTILITIES

JACKPOT

This is it, the ultimate Fruit Machine for the VIC with nudge, hold and respin 100% machine code. "Overall Jackpot is a beautifully written simulation giving superb graphics, animation and use of colour. In fact, this program makes Commodore's Fruit Machine cartridge look unbelievably cheap and nasty." Home Computing Weekly No. 20 19/7/83 .... **£5.50**

KWAZY KWAKS

Accuracy and speed are required for this Shooting Gallery, superb use of colour and graphics in this new and challenging game from the author of Jackpot. 100% machine code, joystick or keyboard control **£5.50**

PACMANIA

Choose your own game from the following options — difficulty 1-3 speed 1-3, size of maze 1-3, visible or invisible maze, still or moving power pills, define your own key controls, any combination, if this is your type of game, then this is the one for you, for the 3K expand VIC **£5.50**

BUGSY (Joystick Only)

This is a Minefield with a difference as you step on the stones whilst collecting purple boxes which give your both time and points, they disappear from beneath your feet. DO NOT DESPAIR! "BUGSY" will randomly replace the stones but avoid bumping into him or its sudden death! An original compulsive and challenging game **£5.50**

MINI-ROULETTE — PONTOON — HI-LOW

Three great games of chance for the VIC, try to beat the computer, hours of fun, full colour, sound effects and tunes **£5.50**

DATABASE — create your own files and records on tape **£7.50**

SUPER BANK MANAGER — A full feature version any memory size, but needs 3K expansion **£7.50**

COMMODORE 64
GAMES AND UTILITIES

JACKPOT 64

At last its here, specially written for the 64, by the author of "JACKPOT" the ultimate Fruit Machine program for the VIC **£5.50**

(available from 1st Nov)

WESTMINSTER

A game for up to four players, can you lead the party of your choice and win the general election, you tour the 60 constituencies (seats) buying votes, when you can, (just like the real thing), this must be one of the first board type games specially written for the computer. Why play on your own, have fun with your family and friends playing WESTMINSTER **£5.50**

RED ALERT

A game for 1-4 players, with sound and graphics. Make money in casinos, commit robbery, hide from the police, hire secret agents, (some of whom can be treacherous), negotiate for weapons, find and attack the secret rocket base to launch the missile, and watch the havoc and destruction caused. There's no turning back from "RED ALERT" **£5.50**

WH££LER DEALER

A game for two to twenty players, become a tycoon of the motor trade, you must obtain gearboxes, tyres and engines to produce cars for sale. Form syndicates, buy and exchange parts, buy dealerships, but be careful, you may become bankrupt and have to liquidate, find out what you are made of, have you got what it takes to become a WH££LER DEALER **£5.50**

LUNAR RESCUE

Our new version, avoid the asteroid belt, to rescue the stranded scientists, then fight your way back to the mother ship, fast reactions are required to safely land and dock your lunar module **£5.50**

PURCHASE LEDGER

Easy to use, single entry, handles 400 invoices per month,
Gross/Nett purchases, V.A.T. **£14.50**
Disk Version **£17.00**
(available from 1st Nov)

SALES LEDGER — As Above **£14.50**
Disk Version **£17.00**

SALES JOURNAL
As above £14.50
Disk version £17.00

CHIPMON — Contains a one- or two- pass Assember, Disassembler and Monitor. A programming aid for the development of machine code programs and routines on the CBM 64 .... **£12.50**

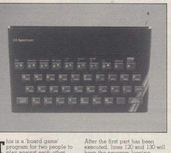
Other software available for the VIC and Commodore 64, send for free brochure, including RABBIT SOFTWARE at **£5.50 each.** Send Cheques/PO's to:

MR CHIP SOFTWARE
Dept **PS, 1 NEVILLE PLACE, LLANDUDNO, GWYNEDD LL30 3BL.**
Tel: 0492 49747
Wanted: High quality software, of all types, for export and UK distribution.
All programs now available on disc please allow £2.50 extra.
DEALER ENQUIRIES WELCOME

Colin Gooch

TOURIST TRAP

Get the holiday horrors in this spectacular board game for your 48K Spectrum.



This is a 'board game' program for two people to play against each other, rather than one person against the computer. It is a fairly large program using a major portion of the 48K RAM available on the larger Spectrum. This means that SAVEing and LOADing will take quite some time. The playing of one game usually takes something like 40 minutes.

You can consider the program as operating in four main parts:

1) The initialisation of values and graphics, the printing of the instructions and then the playing board display.
2) The dice calling routine to decide your throw.
3) The moving of your piece and the diversion if required at that new position onto a subroutine which will decide your score for that throw.
4) The end game routine.

After the first part has been executed, lines 120 and 130 will keep the program looping through the second and third parts, automatically changing the players, until the end game condition is fulfilled to break out into the fourth part.

GOING FOURTH

The four sections operate as follows:
Part One Lines up to 100 call all the initialisation routines. Line 9000 allows the instructions to be displayed while the user-defined graphics are set up. These are the two playing pieces, plus a number of odd shaped 'blobs' used to create pictures in the course of play. The board printout begins at line 8000. The PLOT/DRAW facility draws out a playing track around the outside of the display area allowing the centre to remain clear for the various graphics routines. The playing squares are numbered using different combinations of INK, PAPER and BRIGHTness, these being controlled by the DATA statements at line 8000. It is the attributes on these squares that will decide the 'fate' of a player landing on them. IT is *most* important that these are not altered or you will not be sent to the correct subroutines.
Part Two The dice routine is a straightforward affair. The variables for the 'pips' are in line 8300 onward and it is printed out as required by GOSUB 8425. To roll the dice you press Enter and keep it pressed until you think you have the number you want. Removing the pressure will stop the dice.
Part Three Moving the pieces around the board is more complicated. It is done by a series of FOR . . .NEXT statements. These take the value of the dice thrown as the second control value in the statement and move the piece along that number of squares. If that line runs out of squares then the next FOR . . .NEXT loop will use up the surplus. (I'm usually fond of conditional statements, but in this case found this system operated in a speedier manner.)

On arriving at a square, the attributes of the number of that square is read. If it is 104 then that turn ends, line 6660 changes over the player and your opponent will have their turn. If it is anything other than 104 then the value is converted to a GOSUB line and off you go to see what fate has in store.

The bulk of the program consists of these subroutines which I will let you discover for yourself. The operation of these routines is fairly straightforward. There are nice things that will happen, not so nice things and, of course, some disasters — all of which lose or gain you points, lose you money and use up your petrol. If you run out of petrol you will have to pay for a rescue and if you are on the way to a location, you will lose the rest of that turn. The game ends when one person runs out of money . . . but the winner is the one with the most points, so towards the end of a game tactics can become important!

Part Four Brief and simple. Once one person is out of money, line 130 fails to maintain the loop and the 'game over' signal shows. The final scores are then displayed. The score display routine is at line 8600 and is used at intervals throughout the game. Line 8610 ensures that petrol and money are only displayed to two decimal places; without this line you'll find that as the game progresses you end up with petrol designated in scientific notation!

EASY TO ENTER

Debugging a program of this size can be a bit daunting and so I would suggest it is entered in sections. Enter the program as listed but omitting all the subroutines from line 1000 to 6490. This will allow you to get the board display and the piece moving routines all in working order. To do this enter a temporary line, 6480 Return. RUNning the program will then result in everything happening except any of the scoring subroutines. The subroutines can then be entered one at a time and tested out. There is no need to keep going round the board to test these, just let the Spectrum print out the board, then break into the program and type in GOTO (the appropriate routine) and all will be well. (The 'Press enter to continue' uses the form 'INPUT:LINE Z$' which means that the usual Break and Stop keys will not work. Never fear. Use Caps Shift and '6' and you will get a 'stop in input' message.) The 'out of petrol' routine is the most difficult to test and may best be left until you are happy with the others.

Most of the graphics are straight off the keyboard. The only user-defined graphics that are vital are those for the two playing pieces and the dots on the dice. The others are not so vital but if you enter and RUN line 9000 to 9140 fairly early on, it is easy to see which keys are required.

WE'RE OFF . . .

When all is entered and you are ready to play you will first of all see the instructions, then the Spectrum will choose at random someone to start. The board is then printed out; the player whose turn it is will be shown on the right-hand margin. If you throw the dice and land on a square that is not blue then some graphics and instructions will appear in the centre of the screen. To continue to play, simply follow all the prompts. To enter in the spirit of it all you must imagine that you are on tour in a car, with a tent, and have the opportunity to visit various places . . . wild life parks . . . castles . . . the sea and so on.

So off you go. Happy Holiday.

VARIABLES

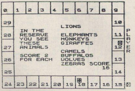
The main variables used are:

C(x)	Cash on hand
D	Value of dice throw.
H(x)	Horizontal position of playing piece.
J(x)	Total score.
LO	Flag indicating turn to be lost.
MI	Miles to travel.
P(x)	Petrol in gallons.
PL	Player.
SUB	Subroutine in use.
TI	Flag to check the number of times the petrol is evaluated.
V(x)	Vertical position of playing piece.
Z$	Line input.

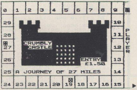
(x) will be one or two according to value of PL.

The subroutines use many more variables, but they are confined within those routines and are not difficult to trace through. There are a number of DIMensioned strings and variables. They are used in several routines and are re-DIMensioned each time that routine is called.

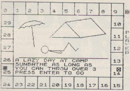
Some sample screen illustrations from the program, Tourist Trap.



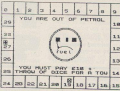
```
                  LIONS
 IN THE
 RESERVE          ELEPHANTS
 YOU SEE          MONKEYS
 THESE            GIRAFFES
 ANIMALS
                  CAMELS
 SCORE 2          BUFFALOS
 FOR EACH         WOLVES
                  ZEBRAS SCORE
                              16
```



```
A JOURNEY OF 27 MILES
```

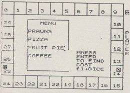

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
PLAYER

A LAZY DAY AT CAMP
SUNBATHE AS LONG AS
YOU CAN THROW OVER 3
PRESS ENTER TO GO


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
PLAYER
fuel

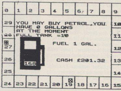
YOU ARE OUT OF PETROL

YOU MUST PAY £10 +
THROW OF DICE FOR A TOW


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
PLAYER

MENU
PRAWNS
PIZZA
FRUIT PIE
COFFEE

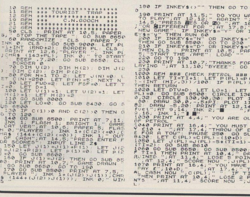
PRESS
ENTER
TO FIND
COST
£1+DICE


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
PLAYER
168

YOU MAY BUY PETROL,YOU
HAVE 0 GALLONS
AT THE MOMENT
FULL TANK =10

FUEL 1 GAL.

CASH £201.32



```
  10 REM ***********************
  11 REM *** TOURIST  TRAP ***
  12 REM ***********************
  19 REM ***   C.N.GOOCH   ***
  21 REM ***********************
  22 REM ***********************
  50 CLS : PRINT AT 10,5; PAPER
5;" STOP THE TAPE ": GO SUB 8650
  60 RANDOMIZE : CLS : PAPER 7:
INK 0: CLS : GO SUB 9000: LET PL
=1+INT (RND*2): BORDER PL: CLS :
 PRINT AT 12,6;"PLAYER "; INK PL
;CHR$ (143+PL); INK 0;" TO START
": BEEP .7,20: GO SUB 8650: CLS
: BORDER 0
  70 DIM V(2): DIM H(2): DIM J(2
): DIM C(2): DIM P(2)
  80 FOR N=1 TO 2: LET J(N)=0: L
ET C(N)=250: LET P(N)=5: NEXT N
  90 LET FWD=2: LET FSF=0: LET T
I=0: LET D=1
 100 LET V(1)=1: LET V(2)=1: LET
 H(1)=1: LET H(2)=2
 110 GO SUB 3000
 120 LET LO=0: GO SUB 8430: GO S
UB 6500
 130 IF C(1)>0 AND C(2)>0 THEN G
O TO 120
 140 GO SUB 8500: PRINT AT 7,11;
 INK 1; FLASH 1; BRIGHT 1;" GAME
 CAT VER";AT 10,6; PAPER 5; FLAS
H 0;"PLAYER "; INK 1+(C(2)<=0);C
HR$ (144+(C(2)<=0)); INK 1;" OUT
 OF CASH";AT 14,5;"PRESS ENTER F
OR SCORES": INPUT LINE Z$
 150 LET J(1)=J(1)+P(1): LET J(2
)=J(2)+P(2): LET P(1)=0: LET P(2
)=0: GO SUB 8600:
 160 IF J(1)=J(2) THEN GO SUB 85
00: PRINT AT 10,7;" GAME DRAWN "
: GO SUB 8650: GO TO 190
 170 GO SUB 8500: PRINT AT 7,5;"
 PLAYER "; INK 1+(J(2)>J(1));CHR
$ (144+(J(2)>J(1))); INK 0;" WIN
S "
 180 IF INKEY$<>"" THEN GO TO 16
0
 190 PRINT AT 11,5;" DO YOU WANT
TO PLAY";AT 12,12;" AGAIN? ";AT
14,5;"PRESS YES OR NO ";AT 16,5
;"OR DIFFERENT FOR ";AT 17,8;" A
 NEW GAME": IF INKEY$="Y" OR INK
EY$="y" THEN RUN 60
 200 IF INKEY$="N" OR INKEY$="n"
THEN GO SUB 8500: GO TO 230
 210 IF INKEY$="D" OR INKEY$="d"
THEN CLS : PRINT AT 10,5; PAPER
5;"START THE ;APE": LOAD ""
 220 GO TO 190
 230 PRINT AT 10,7;"THANKS FOR P
LAYING ";AT 11,10;"BYEEE": GO TO
 230
1000 REM ### CHECK PETROL ###
1010 LET TI=TI+1: LET P(PL)=P(PL
)-(INT (MI/.3))/100: IF P(PL)>0
THEN RETURN
1020 LET DTV=D: LET LO=1: LET P(
PL)=0: GO SUB 8500: CIRCLE 130,9
0,32: CIRCLE 130,90,30: PLOT 115
,80: DRAW 30,0,-.5*PI: PLOT 117,
82: DRAW -5,20: PRINT AT 12,14;"
fuel";AT 9,14; INK 2;"█    ";AT
9,16; INK 1;"▌█"
1030 PRINT AT 4,4;" YOU ARE OUT
OF PETROL "
1040 PRINT AT 16,4;" YOU MUST PA
Y £10 +  ";AT 17,4;"THROW OF DIC
E FOR A TOW": PAUSE 250: GO SUB
8430: LET C(PL)=C(PL)-(10+D):
1050 LET J(PL)=J(PL)-5*(TI=1)-2*
(TI=2): GO SUB 8610
1060 GO SUB 8650: GO SUB 8500: I
F TI=1 THEN PRINT AT 10,4;" DAY
RUINED ";AT 11,4;" LOSE 5 POINTS
 ";AT 12,4;" SCORE NOW ";J(PL)
1070 PRINT AT 16,4;" YOU HAVE TO
 PAY £";10+D;"   ";AT 17,4;" TOT
AL CASH NOW ";C(PL);" ": IF TI=2
 THEN PRINT AT 10,4;" LOSE 2 POI
NTS  ";AT 11,4;" SCORE NOW ";J(P
L)
```

```
1080 GO SUB 8650
1090 GO SUB 8500: GO SUB 4360: L
ET D=DTV: RETURN
2090 REM ### RESTAURANT ###
2100 GO SUB 8650: GO SUB 8500: P
RINT AT 5,5;" GO OUT FOR A MEAL
";AT 6,10;" INSTEAD ";AT 10,6; I
NK 7; PAPER 1; BRIGHT 1;" BURPIO
S              ",AT 11,6;"  Italia
n Restaurant": GO SUB 8650
2110 GO SUB 8500: PLOT 55,138: D
RAW 81,0: DRAW 0,-100: DRAW -81,
0: DRAW 0,100
2120 LET SCM=0: DIM T$(8,10)
2130 DATA "OYSTERS","PRAWNS","ME
LON","SOUP"
2140 DATA "FISH","CHICKEN","STEA
K","PIZZA"
2150 DATA "ICE CREAM","GATEAUX",
"FRUIT PIE","PANCAKES"
2160 DATA "SQUASH","TEA","COFFEE
","COKE"
2170 FOR M=10 TO 40 STEP 10: LET
 NN=1+INT (RND*4): RESTORE (2120
+M): FOR N=1 TO NN: READ T$(M/10
): NEXT N: LET SCM=SCM+NN: NEXT
M
2180 FOR N=5 TO 16: PRINT AT N,7
;T$(5): NEXT N: PRINT AT 5,9; PA
PER 3; INK 7; BRIGHT 1;" MENU "
2190 FOR N=7 TO 13 STEP 2: PAUSE
 25: BEEP .02,N: PRINT AT N,7;T$
((N-5)/2): NEXT N
2200 LET T$(6)="YUK !!": LET T$(
7)="NOT BAD": LET T$(8)="YUMMY"
2210 PAUSE 70: PRINT AT 12,17; P
APER 5;"MEAL WAS   ";AT 13,17;T$(
6+(SCM>8)+(SCM>11)): BEEP .4,1-2
0*(SCM<9)+20*(SCM>11): PRINT AT
14,17;"SCORE ";INT (SCM/2): LET
J(PL)=J(PL)+INT (SCM/2)
2220 GO SUB 8650: PRINT AT 12,17
; PAPER 4;"  PRESS   ";AT 13,17;
"  ENTER   ";AT 14,17;"  TO FIND
 ";AT 15,17;"  COST    ";AT 16,1
7;"   £1+DICE "
2230 IF INKEY$<>"" THEN GO TO 22
30
2240 GO SUB 8650: GO SUB 8500: G
O SUB 8430: LET C(PL)=C(PL)-1-D:
 PRINT AT 14,6;" MEAL COST £";1+
D
2250 IF SCM>11 AND (1+D)<3 THEN
PRINT AT 16,6;" GOOD & CHEAP ";A
T 17,6;" BONUS SCORE 2 ": LET J(
PL)=J(PL)+2
2260 RETURN
3940 REM ### CASTLE ###
3950 GO SUB 3960: GO SUB 3990: R
ETURN
3960 PRINT AT 5,5; PAPER 6;"▟▙▟▙
█          ▟▙▟▙▟";AT 6,5;"█████
█          █████";AT 7,5;"
▌          ████"
3970 FOR N=1 TO 8: PRINT AT N+7,
5; PAPER 6;"███████████████████
█": NEXT N: FOR N=11 TO 15: PRI
NT AT N,14; PAPER 7;"▟▙▟▙▟": NEX
T N
3980 PRINT AT 10,6; PAPER 2; INK
 7; BRIGHT 1;"CRUMBLY";AT 11,7;"
CASTLE": RETURN
3990 LET MI=15+INT (RND*15): PRI
NT AT 17,4; PAPER 5;"A JOURNEY O
F ";MI;" MILES";AT 14,20; PAPER
4;"ENTRY";AT 15,20;" £1.50 "
4000 INPUT "DO YOU WANT TO GO YE
S OR NO?"; LINE Z$: IF Z$="N" OR
 Z$="n" THEN RETURN
4010 GO SUB 1000: IF LO=1 THEN L
ET TI=0: RETURN

4015 IF 1+INT (RND*14)=1 THEN PA
USE 50: PRINT AT 14,20;"CLOSED",
AT 15,20;"TODAY  ": LET J(PL)=J(
PL)-5: PRINT AT 16,4; PAPER 4;"D
ISAPPOINTED!",AT 17,4;" LOSE 5 P
OINTS          ": FOR N=10 TO 1 ST
EP -1: BEEP .3,N: NEXT N: GO SUB
 8650: GO SUB 1000: LET TI=0: GO
 TO 8600: RETURN
4020 LET C(PL)=C(PL)-1.5: RESTOR
E 4030: DIM H$(6,11)
4030 DATA "BORING","VERY DULL","
BEARABLE","FASCINATING","EXCITIN
G","FABULOUS"
4040 FOR N=1 TO 6: BEEP .1,N: RE
AD Q$: LET H$(N)=Q$: NEXT N: PRI
NT AT 4,4; PAPER 5;"THROW DICE T
O SEE IF IT";AT 5,4;"IS INTEREST
ING"
4050 GO SUB 8650: FOR N=1 TO 200
: NEXT N: IF INKEY$<>"" THEN GO
TO 4050
4060 GO SUB 8500: GO SUB 8430: F
OR n=1 TO 300: NEXT n: GO SUB 39
60: PRINT AT 4,4; PAPER 4;"IT IS
 ";H$(D);AT 5,4;" SCORE ";D: LET
 J(PL)=J(PL)+D: LET FC=1: GO TO
4080
4070 GO SUB 8650: PRINT AT 4,4;
PAPER 6;"
;AT 5,4;"                  ": RETURN
4080 IF RND<.2 THEN GO SUB 4070:
 PRINT AT 4,4;"YOU SEE THE GHAST
LY     ": PAUSE 50: PRINT AT 5,4;
 FLASH 1; PAPER 3;" GHOST "; FLA
SH 0; PAPER 6;"  SCORE 5         "
: GO SUB 4130: LET J(PL)=J(PL)+5
: LET FC=0
4090 IF RND<.3 THEN GO SUB 4070:
 PRINT AT 4,4;"GOOD PHOTOS FROM
TOWER ";AT 5,4;"SCORE 3
        ": LET J(PL)=J(PL)+3: LET
 FC=0
4095 IF RND<.25 THEN GO SUB 4070
: PRINT AT 4,4; PAPER 3; INK 7;
BRIGHT 1;"YOU GET LOST IN THE
 ";AT 5,4;"DUNGEON, LOSE 4 POIN
TS": LET J(PL)=J(PL)-4: LET FC=0
4100 IF RND<.3 THEN GO SUB 4070
: PRINT AT 4,4; PAPER 4;" SMASHIN
G CREAM TEA !  ";AT 5,4;"SCORE 4
                ": LET J(PL)=J(P
L)+4: LET FC=0
4110 IF FC THEN GO TO 4080
4120 GO SUB 8650: GO SUB 8500: G
O SUB 1000: LET TI=0: GO SUB 860
0: RETURN
4130 FOR N=1 TO 10
4140 PRINT AT 10,20; OVER 1; PAP
ER 7; INK 0;"E"
4145 PRINT AT 11,20; OVER 1; PAP
ER 7; INK 0;"HI"
4150 PRINT AT 12,20; OVER 1; PAP
ER 7; INK 0;"LL"
4160 PAUSE 20: NEXT N: RETURN
4340 REM ### BUY PETROL ###
4350 IF P(PL)>=7 OR RND<.5 OR LO
 THEN GO TO 4497
4360 GO SUB 8610: IF P(PL)<=0 AN
D C(PL)<=0 THEN GO TO 140
4365 PRINT AT 5,4;"YOU MAY BUY P
ETROL,YOU ";AT 6,4;"HAVE ";P(PL)
;" GALLONS  ";AT 7,4;"AT THE MOM
ENT ";AT 8,4;"FULL TANK =10 ";AT
 14,13;" CASH £";C(PL);" "
4370 DIM H$(5,6)
4380 LET H$(1)="█████ "
4390 LET H$(2)="█▀▀▀█▖"
4400 LET H$(3)="█▄▄▄██"
4410 LET H$(4)="████▐█"
4420 LET H$(5)="████▐█"
4430 DATA 1,2,3,4,4,4,5,1

4440 RESTORE 4430: FOR N=1 TO 8
: READ A: PRINT AT 8+N,5,H$(A): N
EXT N: LET PR=160+INT (RND*10):
PRINT AT 15,6,PR
4445 INPUT "ENTER GALLONS TO BE
BOUGHT"; LINE R$: IF CODE R$<48
OR CODE R$>58 THEN GO TO 4445
4450 LET GA=VAL R$: IF GA*(PR/10
0)>C(PL) THEN LET GA=C(PL)/(PR/1
00): LET P(PL)=P(PL)+GA: LET C(P
L)=0: GO SUB 8610: PRINT AT 9,13
;"CASH FOR ONLY";AT 10,13;(INT G
A*100)/100;" GAL.";AT 14,20;"
  ": GO TO 4470
4454 IF P(PL)+GA>10 THEN GO TO 4
445
4455 IF P(PL)+GA<=10 THEN LET P(
PL)=P(PL)+GA: LET C(PL)=C(PL)-GA
*(PR/100)
4465 GO SUB 8610: PRINT AT 10,13
;"FUEL ";P(PL);" GAL.";AT 14,13;
" CASH £";C(PL)
4470 GO SUB 8650: GO SUB 8500
4490 IF LO OR RND>.5 THEN RETURN

4495 REM ##### SITE FEES #####
4497 IF FSF=1 THEN GO TO 4550
4500 FOR N=5 TO 7: PRINT AT N,6;
```

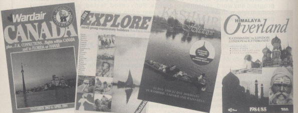
```
"                  ": NEXT N: PR
INT AT 6,7; PAPER 4;" A CAMPING
SITE "
4510 LET SF=3+INT (RND*3): PRINT
 AT 8,4;"YOU MUST PAY SITE FEES
";AT 9,4;"  PRESS ENTER TO FIND
 ";AT 10,13;"  COST  ";AT 13,4;"
COST WILL BE £";SF;" TIMES ";AT
14,4;" THROW OF DICE           "
4515 PAUSE 0: BEEP .2,20: FOR N=
1 TO 150: NEXT N: IF INKEY$<>""
THEN GO TO 4500
4520 GO SUB 8500: GO SUB 8430: P
RINT AT 11,4;"  COST IS  £";SF*D
;"           ": LET C(PL)=C(PL)-(SF
*D)
4530 DIM N$(3,8): LET N$(1)="TER
RIBLE": LET N$(2)="GOOD": LET N$
(3)="PRETTY": LET ST=1+INT (RND*
3)
4540 PAUSE 100: LET J(PL)=J(PL)+
(ST=3)+(ST=2 OR ST=3)-(ST=1): PR
INT AT 15,4;"IT IS A ";N$(ST);"
 SITE ";AT 16,4;"SCORE ";(ST=3)+
(ST=2 OR ST=3)-(ST=1): LET FSF=1
: GO SUB 8650: GO TO 8600
4550 GO SUB 4555: GO TO 4590
4555 GO SUB 8590: FOR N=8 TO 11
 PRINT AT N,5; INK 3;"██████████
███████████": NEXT N: FOR N=9 TO
 10: PRINT AT N,7; INK 1;"▀▄▀▄▀▄
▀▄▀▄▀▄▀▄▀▄": NEXT N
4560 PRINT AT 10,8; PAPER 7; INK
 2; BRIGHT 1;"SWIZWAYS STORES"
4570 FOR N=12 TO 17: PRINT AT N,
5; INK 3;"██           █    ██":
 NEXT N
4580 PRINT AT 17,5; INK 1; PAPER
 3;"▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀": RETURN
4590 PRINT AT 5,5;"YOU NEED PROV
ISIONS": GO SUB 8650
4600 PRINT AT 6,5; PAPER 4;"DICE
 WILL DECIDE WHAT";AT 7,5;"HAPPE
NS IN STORE": GO SUB 8650: GO SU
B 8500: GO SUB 8430
4610 RESTORE 4610: DATA "OUTRAGE
OUS","EXPENSIVE","AVERAGE","FAIR
LY CHEAP","VERY CHEAP","GIVE AWA
Y"
4620 FOR N=1 TO D: READ Q$: NEXT
 N: LET DV=D: PRINT AT 12,4;"THE
 PRICES IN THE SHOP ";AT 13,4;"A
RE ";Q$
4630 PRINT AT 15,5;" SCORE ";D:
LET J(PL)=J(PL)+D: GO SUB 8650
4640 GO SUB 8500: LET PR=12+INT
(RND*5): PRINT AT 12,4;"YOU EXPE
CT TO SPEND £";PR;AT 13,4;"DICE
WILL TELL YOU ";AT 14,4;"SIZE OF
 BILL ": PAUSE 90: GO SUB 8430
4650 LET PP=5+D*3: PRINT AT 13,4
;"YOU ACTUALLY SPEND £";PP;AT 14
,4; PAPER 6;"                   "
4560 LET C(PL)=C(PL)-PP
4670 IF PP<PR-2 THEN PRINT AT 16
,4;"YOU SAVE MONEY ": IF DV=1 OR
 DV=2 THEN PRINT AT 16,19;"ALTHO
UGH";AT 17,4;"IT IS ";Q$;"     "
4680 IF PP<PR-2 THEN PRINT AT 17
,20;"SCORE ";3+3*(DV<3): LET J(P
L)=J(PL)+3+3*(DV<3): GO TO 4720
4690 IF PP>PR+2 THEN PRINT AT 16
,4;"YOU OVER SPEND ": IF DV>4 TH
EN PRINT AT 16,19;"ALTHOUGH";AT
17,4;"IT'S ";Q$;"     "
4700 IF PP>PR+2 THEN PRINT AT 17
,20;"SCORE";-3-3*(DV>4): LET J(P
L)=J(PL)-3-3*(DV>4): GO TO 4720
4710 PRINT AT 16,4;"YOU ARE CLOS
E TO YOUR ";AT 17,4;"BUDGET. SCO
RE 2": LET J(PL)=J(PL)+2
4720 LET FSF=0: GO SUB 8650: GO
TO 8600
4740 REM #### LAZY DAY ####
4750 LET SCD=0: PLOT 39,120: DRA
W 50,10,-.5*PI: FOR N=1 TO 5: DR
AW -10,-2,.5*PI: NEXT N: PLOT 64
,125: DRAW 5,-40: PLOT 39,64: DR
AW 175,0: PLOT 100,104: DRAW 30,
30: DRAW 30,-30: PLOT 130,134: D
RAW 50,8: DRAW 30,-30: DRAW -50,
-8: DRAW -56,0
4760 CIRCLE 90,80,7: DRAW 30,0:
DRAW 10,16: DRAW 14,-16: DRAW 5,
5: PLOT 100,77: DRAW 22,-4: PRIN
T AT 14,4;"A LAZY DAY AT CAMP"
4770 PRINT AT 15,4;"SUNBATHE AS
LONG AS";AT 16,4;"YOU CAN THROW
OVER 3";AT 17,4; PAPER 4;"PRESS
ENTER TO GO": PAUSE 0: GO SUB 84
30: PRINT AT 10,20; FLASH 1;"AG
AIN": LET J(PL)=J(PL)+D: LET SCD
=SCD+D: IF D>3 THEN GO TO 4770
4780 PRINT AT 10,12; PAPER 5;"SU
N'S GONE IN": FOR N=0 TO -10 STE
P -1: BEEP .05,N: NEXT N: IF SCD
<=8 THEN GO SUB 2000
4790 GO SUB 8650: GO SUB 8500: G
O TO 8600
4795 REM #### GOLF ####
4800 GO SUB 4810: GO SUB 4860: R
ETURN
4810 FOR N=4 TO 9: PRINT AT N,4;
 PAPER 5;"
  ": NEXT N
4820 FOR N=10 TO 17: PRINT AT N,
4; PAPER 4;"
  ": NEXT N
4830 PRINT AT 12,6; PAPER 4; INK
 6;"DEG"
4835 PRINT AT 13,6; PAPER 4; INK
 6;"H█I"
4840 PRINT AT 14,6; PAPER 4; INK
 6;"JKM"
4850 PLOT 180,120: DRAW 0,-24: P
RINT AT 7,23;"1": RETURN
4860 PRINT AT 6,6;" A DAY AT ";A
T 7,6;" THE GOLF CLUB ": PAUSE 7
5: PRINT AT 12,12;" YOU PLAY YOU
R ";AT 13,12;" OPPONENT IN ";AT
14,12;" THIS GAME "
4870 GO SUB 8650: GO SUB 4810: P
RINT AT 6,6;"LOSER OF GAME";AT 7
,6;" PAYS FOR";AT 8,6 ;"THE GAME
, COST £10";AT 9,6;"WINNER GETS
10 POINTS"
4880 DIM S(2,18)
4890 GO SUB 8650: GO SUB 8500: P
RINT AT 10,5;"PLAYER "; INK 1;"A
","PLAYER "; INK 2;"B";AT 4,4;"H
OLD ""S"" TO SPEED GAME "
4900 DATA 2,3,4,3,4,2,3,3,4,4,2,
3,5,4,3,2,4,3
4910 LET DR=0: LET GSA=0: LET GS
B=0: RESTORE 4900: LET HOL=1+INT
 (RND*80): FOR N=1 TO 18
4920 READ P: PRINT AT 6,6;"HOLE
NUMBER ";N;AT 7,6;"PAR = ";P+1
4930 LET S(1,N)=P+INT (RND*3)
4940 LET S(2,N)=P+INT (RND*3)
4950 LET GSA=GSA+S(1,N): LET GSB
=GSB+S(2,N)
4960 PRINT AT 12,7;S(1,N);AT 12,
17;S(2,N);AT 15,7;GSA;AT 15,17;G
SB
4970 IF GSA=HOL AND DR=0 THEN PR
INT AT 12,7;"1": FOR M=1 TO 40 S
TEP 4: BEEP .4,M: PRINT AT 17,4;
 FLASH 1;"HOLE IN ONE PLAYER ";
INK 1;"A": NEXT M: LET GSA=GSA-S
(1,N)+1: LET DR=1: PRINT AT 17,4
, PAPER 6;"                    "
4980 IF GSB=HOL AND DR=0 THEN PR
INT AT 12,17;"1": FOR M=1 TO 40
STEP 4: BEEP .4,M: PRINT AT 17,4
; FLASH 1;"HOLE IN ONE PLAYER ";
 INK 2;"B": NEXT M: LET GSB=GSB-
S(2,N)+1: LET DR=2: PRINT AT 17,
4; PAPER 6;"
"
4990 PAUSE 150: BEEP .5,20: NEXT
 N: PAUSE 50: GO SUB 4810
4995 IF GSA=GSB THEN PAUSE 100:
PRINT AT 5,4;"GAME DRAWN. ";AT 6
,4;"SHARE COST AND POINTS": LET
J(1)=J(1)+5: LET J(2)=J(2)+5: LE
T C(1)=C(1)-5: LET C(2)=C(2)-5
5000 IF GSA>GSB THEN LET J(2)=J(
2)+10: LET C(1)=C(1)-10: PRINT A
T 5,4;"PLAYER "; INK 2;"B"; INK
0;" WINS"
5010 IF GSA<GSB THEN LET J(1)=J(
1)+10: LET C(2)=C(2)-10: PRINT A
T 5,4;"PLAYER "; INK 1;"A"; INK
0;" WINS"
5015 IF DR=0 THEN GO TO 5060
5020 IF DR=1 THEN PRINT AT 7,4;"
PLAYER "; INK 1;"A"; INK 0;" HAD
 ";AT 8,4;"A HOLE IN ONE"
5030 IF DR=2 THEN PRINT AT 7,4;"
PLAYER "; INK 2;"B"; INK 0;" HAD
 ";AT 8,4;"A HOLE IN ONE"
```

```
5040 GO SUB 8650: GO SUB 4810: I
F DR>0 THEN PRINT AT 10,10;"A RO
UND AT 19th";AT 11,10;"TEE WILL
COST :";AT 12,10;"£2xTHROW OF DI
CE": PAUSE 200: GO SUB 8430
5050 IF DR>0 THEN LET C(DR)=C(DR
)-2*D: PRINT AT 17,5;"COST £";2*
D
5060 GO SUB 8650
5070 GO TO 8600
5150 REM #### BEACH TRIP ####
5160 LET D=6: GO SUB 5170: GO TO
 5230
5170 LET SUB=5170: FOR N=4 TO 10
: PRINT AT N,4; PAPER 5;"
                  ": NEXT N: FOR N
=11 TO 12: PRINT AT N,4; PAPER 1
;"
                           ": NEXT
 N
5180 IF D>=4 THEN FOR N=1 TO 10:
 INK 6: PAPER 5: CIRCLE 160,120,
N: NEXT N: INK 0: PAPER 7
5190 IF D<=3 THEN PRINT AT 6,18;
 PAPER 5; INK 7;"DEFEFEFG";AT 7,
18;"H██████I";AT 8,18;"H██████I"
;AT 9,18;"JKLKLKLM"
5200 IF D<=2 THEN FOR N=10 TO 16
 STEP 2: FOR M=11 TO 17 STEP 2:
PRINT AT N,18;" ' ' ' ' ";AT M,1
8;" ' ' ' ' ": NEXT M: NEXT N
5210 IF D=1 THEN PLOT 180,90: IN
K 5: BRIGHT 1: DRAW -5,-10: DRAW
 3,0: DRAW -5,-15: INK 0: BRIGHT
 0
5220 RETURN
5230 FOR N=13 TO 17: PRINT AT N,
14; PAPER 7; INK 3;"█        █"
: NEXT N: PRINT AT 9,5;" A TRIP T
O BEACHY BAY";AT 13,14; INK 3;"█
█████████";AT 17,14;"██████████"
;AT 14,15; INK 1;"CAR PARK";AT 1
5,17;"£1";AT 16,15;"A DAY": GO S
UB 8650
5235 FOR N=13 TO 17: PRINT AT N,
14; PAPER 6;"          ": NEXT N
5240 LET MI=20+INT (RND*10): PRI
NT AT 13,5;"YOU MAY GO ON A VISI
T";AT 14,5;" TO THE SEA ";AT 15,
5;" IT IS ";MI;" MILES ": INPUT
" ENTER "; FLASH 1;"Y"; FLASH 0;"
ES OR "; FLASH 1;"N"; FLASH 0;"
O"; LINE Z$: IF Z$="N" THEN RETU
RN
5245 GO SUB 1000: IF LO=1 THEN L
ET TI=0: RETURN
5250 LET C(PL)=C(PL)-1: LET LO=0
: GO SUB 8500: PRINT AT 10,5;" T
HROW DICE TO FIND ";AT 11,5;" O
UT THE WEATHER.   ";AT 12,5;" T
HE HIGHER SCORE   ";AT 13,5;" T
HE BETTER
         "
5260 GO SUB 8430: DIM H$(6,6)
5270 LET H$(1)="STORMY": LET H$(
2)="RAINY": LET H$(3)="CLOUDY":
LET H$(4)="WARM": LET H$(5)="HOT
": LET H$(6)="SUPER"
5280 PAUSE 25: GO SUB 8500: GO S
UB SUB: LET J(PL)=J(PL)+D: PRINT
 AT 10,5;" THE WEATHER IS ": PAU
SE 50: PRINT AT 11,5;H$(D);" SCO
RE ";D
5290 IF SUB<>5170 THEN RETURN
5300 IF D>2 AND RND<.6 THEN GO S
UB 8650: PRINT AT 5,4;"YOU WIN A
";AT 6,4;"SANDCASTLE";AT 7,4;"CO
MPETITION": LET PRI=2+INT (RND*4
): PRINT AT 8,4;"SCORE ";PRI;" P
OINTS": LET J(PL)=J(PL)+PRI
5305 IF D>2 THEN GO TO 5580
5310 PAUSE 70: PRINT AT 5,4;" GO
 ON THE ";AT 6,4;" PIER OUT OF";
AT 7,4;" THE RAIN": GO SUB 8650
5320 GO SUB 8500: FOR N=5 TO 17
STEP 2: PRINT AT N,4+INT N/2; PA
PER 1+INT (RND*5); INK 9; FLASH
INT (RND*2); BRIGHT INT (RND*2);
"PENNY ARCADE": FOR M=1 TO 50 ST
EP 7: BEEP .2-N/100,M: NEXT M: N
EXT N
5330 DIM J$(11,13): PAUSE 30: GO
 SUB 8500: PRINT AT 5,5; PAPER 5
;" FRUIT MACHINE "
5340 LET J$(1)="█████████████"
5345 LET J$(2)="██   ███   ██"
5350 LET J$(4)="██         ██"
5355 LET J$(9)="█████   █████"
5360 LET J$(3)=J$(2): LET J$(5)=
J$(4): LET J$(6)=J$(4): LET J$(7
)=J$(2): LET J$(8)=J$(2): LET J$
(10)=J$(2): LET J$(11)=J$(4)
5370 FOR N=1 TO 11: PRINT AT 6+N
,6; PAPER 7;J$(N): NEXT N: FOR N
=10 TO 15: PRINT AT N,20; PAPER
6;"█": NEXT N: PRINT AT 15,19; P
APER 6;"█"
5380 PRINT AT 6,21; PAPER 4;"POI
NTS";AT 8,23; PAPER 7;" 2 ";AT 9
,23;"--A"
5390 PRINT AT 11,23;" 4 ";AT 12,
23;"-BB";AT 14,23;" 8 ";AT 15,23
;"CCC"
5400 LET WT=0: LET TU=1: LET GO=
3+INT (RND*3): PRINT AT 17,8; PA
PER 2; INK 7; BRIGHT 1;"10p A GO
";AT 10,8; PAPER 1;"YOU HAVE";
AT 11,8;"  ";GO;" GOES ";AT 12,8
;"  ";"  "
5410 INPUT "PRESS ENTER TO PLAY
"; LINE Z$: LET W=0
5420 PRINT AT 7,8; PAPER 3; INK
7; BRIGHT 1;"GO ";AT 7,14;" ";TU
;" "
5430 LET NN=20+INT (RND*10): FOR
 N=1 TO NN: LET FMA=(144+INT (RN
D*3)): LET FMB=(144+INT (RND*3))
: LET FMC=(144+INT (RND*3))
5440 FOR M=10 TO 12: PRINT AT M,
8; PAPER 5;"         ": NEXT M
5450 PRINT AT 11,9; INK 3;CHR$ F
MA;AT 11,12; INK 1;CHR$ FMB;AT 1
1,15; INK 2;CHR$ FMC: NEXT N
5460 IF FMC=144 THEN LET W=2
5470 IF FMB=145 AND FMC=145 THEN
 LET W=4
5480 IF FMA=146 AND FMB=146 AND
FMC=146 THEN LET W=8
5490 PRINT AT 15,12;W: LET TU=TU
+1: LET WT=WT+W: IF TU<>GO+1 THE
N BEEP .5,TU: GO TO 5410
5500 GO SUB 8650: GO SUB 8500: P
RINT AT 6,5;" YOU SCORED ";WT;"
POINTS ";AT 8,5;" IT COST YOU £"
;GO/10: LET J(PL)=J(PL)+WT: LET
C(PL)=C(PL)-(GO/10)
5590 GO SUB 8650: GO SUB 1000: L
ET TI=0: GO TO 8600
5600 REM ##### RAILWAY #####
5610 LET DSL=0: GO SUB 5620: GO
TO 5700
5620 DIM J$(6,17)
5630 LET J$(1)="  █          ████"
5635 LET J$(2)="  █     █    █  █"
5640 LET J$(3)=" ████████████  █"
5645 LET J$(4)="█████████████  █"
5650 LET J$(5)=" ████████████  █"
5655 LET J$(6)="  ████████████████"
5660 FOR N=1 TO 6: PRINT AT 8+N,
5; PAPER 6;J$(N): NEXT N
5670 FOR N=5 TO 13: CIRCLE 80,54
,N: CIRCLE 120,54,N: NEXT N: FOR
 N=50 TO 53: PLOT 75,N: DRAW 56,
0: NEXT N
5680 PRINT AT 17,4; PAPER 6;"███
███████████████████████████"
5690 FOR N=7 TO 17: PRINT AT N,2
6;"█": NEXT N: PRINT AT 8,22; IN
K 2;"█"; INK 7;"█"; INK 2;"██":
RETURN
5700 PRINT AT 5,8;" THE FULL O T
YN ";AT 6,11;" RAILWAY ": GO SUB
 8650: GO SUB 5710: GO TO 5720
5710 FOR N=4 TO 6: PRINT AT N,4;
 PAPER 6;"
": NEXT N: RETURN
5720 LET MI=15+INT (RND*10): PRI
NT AT 5,5;"THE RAILWAY IS ";MI;A
T 6,10;" MILES AWAY";AT 16,18; P
APER 4;"COST £2": INPUT "DO YOU
WANT TO GO YES OR NO"; LINE Z$:
IF Z$="N" OR Z$="n" THEN RETURN
5730 GO SUB 1000: IF LO=1 THEN L
ET TI=0: RETURN
5740 LET C(PL)=C(PL)-2: IF RND>.
6 THEN GO TO 5780
5750 GO SUB 5710: PRINT AT 4,5;"
NO STEAM TRAINS TODAY";AT 5,5;"L
OSE 3 POINTS";AT 16,18; PAPER 6;
"       ": LET J(PL)=J(PL)-3
5760 GO SUB 8650: IF RND>.9 THEN
 GO SUB 5710: PRINT AT 4,5;"NEIT
HER ANY": LET DSL=1
```

```
6770 PRINT AT 5,5,"DIESELS RUNN
NG TODAY";AT 6,5;"SCORE ",-3*(D
L=1)+4*(DSL=0): LET J(PL)=J(PL)
S*(DSL=1)+4*(DSL=0): GO TO 5790
5780 GO SUB 5710: PRINT AT 4,5;"
YOU'RE IN LUCK";AT 5,5;"STEAM TR
AINS RUNNING";AT 6,5;"TODAY. SCO
RE 5": LET J(PL)=J(PL)+5
5790 LET FL=1: IF RND<.8 THEN GO
 TO 5850
5800 GO SUB 8650: GO SUB 5710: P
RINT AT 4,10; PAPER 2; INK 7; FL
ASH 1; BRIGHT 1;" ACCIDENT "
5810 DIM S$(3,7): LET S$(3)="BRO
KEN": LET S$(2)="TWISTED": LET S
$(1)="GRAZED": LET Q=1+INT (RND*
3)
5820 PRINT AT 5,4;"YOU HAVE FALL
EN OFF THE";AT 6,4;"PLATFORM AND
 ";S$(Q);AT 7,10;"YOUR ANKLE": L
ET LG=3+2*(Q=2)+7*(Q=3)
5830 GO SUB 8650: GO SUB 5710: P
RINT AT 5,5;"LOSE ";LG;" POINTS
";AT 7,10; PAPER 6;"           "
: LET J(PL)=J(PL)-LG: LET FL=0
5850 IF RND>.5 THEN GO SUB 8650:
 GO SUB 5710: PRINT AT 5,5;" YOU
 SEE FAMOUS TRAIN ";AT 6,5;" SCO
RE 4 POINTS ": LET J(PL)=J(PL)+4
: LET FL=0
5860 IF RND>.92 THEN GO SUB 8650
: GO SUB 5710: PRINT AT 5,5;" YO
U GET A RIDE ON ";AT 6,5;" THE F
OOTPLATE ";AT 7,5;"SCORE 8 POINT
S ": LET J(PL)=J(PL)+8: LET FL=0
5870 IF FL THEN GO TO 5790
5900 GO SUB 8650: GO SUB 1000: L
ET TI=0: GO SUB 8600: RETURN
5990 REM ##### PARK #####
6000 LET MI=25+INT (RND*10): GO
SUB 6010: GO TO 6080
6010 DIM J$(9,16)
6020 LET J$(1)="
6025 LET J$(2)="
6030 LET J$(3)="
6035 LET J$(4)="
6040 LET J$(5)="
6045 LET J$(6)="
6050 LET J$(7)="
6055 LET J$(8)="
```

```
6060 LET J$(9)=J$(8)
6070 LET SUB=6070: FOR N=1 TO 9:
 PRINT AT N+6,5;J$(N): NEXT N: P
RINT AT 4,5; PAPER 4;"     WILDWO
ODS   ";AT 5,5;"    SAFARI PARK
": RETURN
6080 PRINT AT 7,22, PAPER 5;"ENT
RY";AT 8,22;"£2.50";AT 17,4;"TO
GET THERE ";MI;" MILES"
6090 INPUT "WANT TO GO? YES OR N
O"; LINE Z$: IF Z$="N" OR Z$="n"
 THEN RETURN
6100 GO SUB 1000: IF LO=1 THEN L
ET TI=0: RETURN
6110 LET C(PL)=C(PL)-2.50: GO SU
B 5250
6120 DATA "LIONS","TIGERS","ELEP
HANTS","MONKEYS","GIRAFFES","RHI
NOS","CAMELS","BUFFALOS","WOLVES
","ZEBRAS","GNUS"
6130 LET AN=0: RESTORE 6120: GO
SUB 8650: GO SUB 8500: FOR N=1 T
O 11: READ Q$: IF RND<.3 THEN NE
XT N
6140 PRINT AT 7,5, PAPER 5;"IN T
HE ",AT 8,5;"RESERVE";AT 9,5;"YO
U SEE";AT 10,5;"THESE  ",AT 11,5
;"ANIMALS";AT 13,5;"SCORE 2 ";AT
 14,5;"FOR EACH"
6150 IF N>4 AND FWD<0 AND 1+INT
(RND*50)<=5 THEN GO TO 6300
6160 PAUSE 80: PRINT AT N+5,15;Q
$: LET J(PL)=J(PL)+2: LET AN=AN+
2: PRINT AT 15,22;"SCORE";AT 16,
25;AN: NEXT N: PAUSE 100
6170 LET FL=1: GO SUB 8500: GO S
UB 6070
6180 IF RND>.8 THEN GO SUB 8650:
 PRINT AT 4,5,"YOU SLIP IN SEALI
ON ",AT 5,5;"POOL LOSE 3 POINTS
": LET J(PL)=J(PL)-3: LET FL=0
6190 IF RND>.8 THEN GO SUB 8650:
 PRINT AT 4,5;"HAVE SUPER ICE CR
EAM ",AT 5,5;"SCORE 2 POINTS
": LET J(PL)=J(PL)+2: LET FL=0
6200 IF RND>.7 THEN GO SUB 8650
 PRINT AT 4,5;"GET STUNG BY A WA
SP ";AT 5,5;"LOSE 4 POINTS
": LET J(PL)=J(PL)-4: LET FL=0
6210 IF FL THEN GO TO 6180
6270 GO SUB 8650: GO SUB 1000: L
ET TI=0: LET FWD=FWD-1: GO TO 86
00
6300 FOR N=8 TO 13: PRINT AT N,4
; PAPER 2; INK 7; BRIGHT 1; FLAS
H 1;"                      ": N
EXT N
6310 PRINT AT 10,7; PAPER 1; INK
 7;" D I S A S T E R ": GO SUB
8650
6320 PRINT AT 9,4;" YOU BREAK DO
WN IN THE ";AT 10,4;"       RESE
RVE        ";AT 11,4;" PRESS EN
TER TO SEE HOW";AT 12,4;" MANY P
OINTS YOU LOSE  "
6330 PAUSE 0: BEEP .05,0: FOR N=
1 TO 90: NEXT N: IF INKEY$<>"" T
HEN GO TO 6330
6340 GO SUB 8500: PRINT AT 15,8;
"LOSE 3+ DICE ": GO SUB 8430: PR
INT AT 15,8; PAPER 4;" LOSE ";3+
D;" POINTS": LET J(PL)=J(PL)-D-3
6350 GO SUB 8650: GO SUB 1000: L
ET TI=0: LET FWD=4: GO TO 8600
6490 REM #### MOVE PIECES ####
6500 IF INKEY$<>"" THEN GO TO 65
00
6510 PRINT AT V(PL),H(PL);" ": B
EEP .3,D: LET P(PL)=P(PL)-.25: I
F P(PL)<=.009 THEN GO SUB 1020:
LET LO=0
6520 FOR N=1 TO D: IF V(PL)=1 TH
EN LET H(PL)=H(PL)+3: IF H(PL)>2
8+(PL=2) THEN LET V(PL)=V(PL)+(H
(PL)-(28+(PL=2))): LET H(PL)=28+
(PL=2): LET D=D-N: GO TO 6540
6530 NEXT N
6540 FOR N=1 TO D
6550 IF V(PL)>1 AND H(PL)=28+(PL
=2) THEN LET V(PL)=V(PL)+3: IF V
(PL)>19 THEN LET H(PL)=H(PL)-(V(
PL)-19): LET V(PL)=19: LET D=D-N
: GO TO 6570
6560 NEXT N
6570 FOR N=1 TO D
6580 IF V(PL)=19 AND H(PL)<28+(P
L=2) THEN LET H(PL)=H(PL)-3: IF
H(PL)<1+(PL=2) THEN LET V(PL)=V(
PL)+(-1+H(PL)-(PL=2)): LET H(PL)
=1+(PL=2): LET D=D-N: GO TO 6600
6590 NEXT N
6600 FOR N=1 TO D
6610 IF V(PL)<18 AND H(PL)=1+(PL
=2) THEN LET V(PL)=V(PL)-3: IF V
(PL)<1 THEN LET H(PL)=H(PL)+(1-V
(PL)): LET V(PL)=1: LET D=D-N: G
O TO 6500
6620 NEXT N
6630 PRINT AT V(PL),H(PL); INK P
L,CHR$ (144+(PL=2))
6650 LET SUB=ATTR (V(PL)+1,H(PL)
): IF SUB<>104 THEN GO SUB 8500:
 GO SUB SUB*50: LET LO=0
6660 LET PL=PL+(PL=1)-(PL=2): GO
 SUB 8500: GO SUB 8510: PRINT AT
 2+18*(PL=2),31; INK 1+(PL=2);CH
R$ (144+(PL=2)): RETURN

7990 REM ### SET UP BOARD ####
8000 DATA 4,26,148,172,4,28,244,
220,5,0,5,0,5,0,4,0,5,0,5,0,2,7,
5,0,5,0,4,7,5,0,5,0,5,0,5,0,2,7,
5,0,5,0,6,0,5,0,5,0,5,0,2,7,1,7,
5,0,3,7,5,0,5,0,7,0,2,7,5,0
8110 RESTORE 8000: FOR N=1 TO 4:
 READ A: PLOT 4,A: DRAW 240,0: N
EXT N
8120 FOR Y=4 TO 172 STEP 24
8130 PLOT 4,Y: DRAW 25,0: PLOT 2
20,Y: DRAW 24,0: NEXT Y
8140 FOR N=1 TO 4: READ A: PLOT
A,4: DRAW 0,168: NEXT N
8150 FOR X=4 TO 248 STEP 24
8160 PLOT X,4: DRAW 0,24: PLOT X
,148: DRAW 0,24: NEXT X
8170 GO SUB 8500
8180 LET NA=0: LET NB=34: LET NC
=29: LET ND=5
8190 FOR L=1 TO 2: FOR N=1 TO 28
 STEP 3: IF L=1 THEN READ A: REA
D B: PRINT AT 2,N; PAPER A; BRIG
HT 1; INK B;NA;" "
8200 IF L=2 THEN READ A: READ B:
```

```
PRINT AT 20,N; PAPER A; BRIGHT
1; INK B;NB
8210 LET NA=NA+1: LET NB=NB-1: N
EXT N: NEXT L
8220 FOR L=1 TO 2: FOR N=5 TO 17
 STEP 3: IF L=1 THEN READ A: REA
D B: PRINT AT N,1; BRIGHT 1; PAP
ER A, INK B;NC
8230 IF L=2 THEN READ A: READ B:
 PRINT AT N,26; PAPER A; BRIGHT
1; INK B;ND
8240 LET NC=NC-1: LET ND=ND+1: N
EXT N: NEXT L
8250 GO SUB 8510
8260 PRINT AT 1,1; INK 1;"A";AT
1,2; INK 2;"B"
8270 PRINT AT 2+17*(PL=2),31; IN
K 1+(PL=2);CHR$ (144+(PL=2))
8300 REM #### DICE ####
8310 DIM A$(6,3): DIM B$(6,3): D
IM C$(6,3)
8320 LET A$(1)="   "
8330 LET B$(1)=" C "
8340 LET C$(1)=A$(1)
8350 LET A$(2)="C  "
8360 LET B$(2)=A$(1)
8370 LET C$(2)="  C"
8380 LET A$(3)=A$(2): LET B$(3)=
B$(1): LET C$(3)=C$(2)
8390 LET A$(4)="C C": LET B$(4)=
A$(1): LET C$(4)=A$(4)
8400 LET A$(5)=A$(4): LET B$(5)=
B$(1): LET C$(5)=A$(4)
8410 LET A$(6)="CCC": LET B$(6)=
A$(1): LET C$(6)=A$(6)
8420 LET D$="▛▀▀▀▜": LET E$="▙▄▄
▄▟": RETURN
8425 REM ###PRINT DICE ###
8430 LET D=1+INT (RND*6): PRINT
AT 5,5;D$;AT 9,5;E$
8440 FOR N=6 TO 8: PRINT AT N,5;
"▌";AT N,9;"▐": NEXT N
8450 PRINT AT 6,12;"PRESS ENTER
TO";AT 7,12;" ""THROW"" DICE ";A
T 8,11;" RELEASE TO STOP": INPUT
 LINE Z$
8460 IF D=7 THEN LET D=1
8470 PRINT AT 6,6;A$(D);AT 7,6;B
$(D);AT 8,6;C$(D)
8480 IF INKEY$="" THEN RETURN
8490 LET D=D+1: GO TO 8460
8495 REM ### BLANK BOARD ###
8500 FOR Y=4 TO 17: PRINT AT Y,4
; PAPER 6;"
": NEXT Y: RETURN
8505 REM ### BLANK LAST COL ###
8510 FOR N=0 TO 21: PRINT AT N,3
1; PAPER 0;" ": NEXT N
8520 DATA "P","L","A","Y","E","R
"
8530 RESTORE 8520: FOR N=7 TO 12
: READ R$: PRINT AT N,31; BRIGHT
 1; PAPER 0; INK 6;R$: BEEP .005
,N: NEXT N: RETURN
8590 REM ### DISPLAY SCORE ###
8600 GO SUB 8610: GO TO 8620
8610 LET P(PL)=(INT (P(PL)*100))
/100: LET C(PL)=(INT (C(PL)*100)
/100): RETURN
8620 GO SUB 8500: PRINT AT 6,6;"
PLAYER "; INK 1;"A ";AT 8,6; INK
 0;"SCORE NOW ";J(1);AT 9,6;"CAS
H NOW £";C(1);AT 10,6;"PETROL LE
FT ";P(1);AT 12,6;"PLAYER "; INK
 2;"B ";AT 14,6; INK 0;"SCORE NO
W ";J(2);AT 15,6;"CASH NOW £";C(
2);AT 16,6;"PETROL LEFT ";P(2)
8650 IF INKEY$<>"" THEN GO TO 86
50
8660 INPUT "PRESS ENTER TO CONTI
NUE"; LINE Z$: BEEP .2,20: RETUR
N
9000 REM ###INSTRUCT ###
9010 PRINT AT 1,8; PAPER 2; INK
7; BRIGHT 1;" TOURIST TRAP ": GO
 SUB 9100: BORDER 4
9020 PRINT AT 3,2;" YOUR OBJECT
IS TO SCORE AS "'" MANY POINTS A
S POSSIBLE BEFORE"'" ONE OF THE
PLAYERS RUNS OUT"'" OF MONEY."'"
    YOU MOVE ROUND THE BOARD TO"
'" THE THROW OF A DICE."'"   YOU
 THROW RIGHT THROUGH YOUR"'" TUR
N EVEN IF DICE APPLIES TO"'" YOU
R OPPONENT."
9030 PRINT AT 13,2;" IT IS AS WE
LL TO BUY PETROL"'" WHEN YOU CAN
. ANY LEFT AT"'" THE END OF THE
GAME WILL BE "'" CREDITED AS POI
NTS"
9040 PRINT AT 18,2;"PLAYER ONE H
AS SYMBOL "; INK 1;"A"; INK 0;'"
   PLAYER TWO HAS SYMBOL "; INK
2;"B"
9050 GO SUB 8650: RETURN
9100 REM ### GRAPHICS ###
9110 DATA 255,153,153,255,255,15
3,153,255,255,195,165,153,153,16
5,195,255,0,0,60,60,60,60,0,0
9120 DATA 0,0,0,0,5,15,7,15,0,12
,28,125,255,255,255,255,0,0,0,0,
193,231,239,255,0,0,0,0,192,192,
224,240,15,15,31,63,127,63,63,15
,240,224,252,252,248,240,224,240
,15,15,3,1,0,0,0,0,255,255,255,2
55,124,60,24,16,255,255,223,135,
5,4,0,0,240,224,224,128,0,0,0,0
9130 RESTORE 9110: FOR N=0 TO 12
: FOR M=0 TO 7: BORDER M: BEEP .
005,N
9140 READ A: POKE USR CHR$ (97+N
)+M,A: NEXT M: NEXT N: RETURN
9900 SAVE "TOUR" LINE 10
9910 PRINT AT 10,2; PAPER 5;"REW
IND TAPE.CHANGE PLUGS FOR"'" VER
IFY. START THE TAPE "'" PRESS EN
TER TO VERIFY "
9920 VERIFY "TOUR"
```



UPGRADE YOUR ZX SPECTRUM NOW!

The "CHEETAH" 32K Rampack simply plugs into the user port at the rear of your computer and increases the memory instantly to 48K.

- Fully compatible with all accessories via rear edge connector.
- No need to open computer and invalidate guarantee.
- Why send your computer away and wait weeks for upgrade.
- Fully cased, tested and guaranteed.

Why wait any longer?

ONLY **£39.95** inc. VAT and p&p

32K CHEETAH
SWEET TALKER

NOW MAKE YOUR SPECTRUM AND ZX81 TALK!

The CHEETAH "Sweet Talker" just plugs into the back of the computer using the existing power supply. Based on an allophone system you can easily program any word, sentence or phrase. Fully cased, tested, guaranteed and compatible with all accessories via rear edge connectors. Complete with demonstration cassette and full instructions. No more lonely nights!

Simply incredible at **£29.75**

Also available: Please quote when ordering whether **Spectrum** or **ZX** owner

16K Rampack for ZX81 ...... **£19.75** 64K Rampack for ZX81 ...... **£44.75**

Prices include VAT, Postage and Packing. Delivery normally 14 days. Export orders at no extra cost. Dealer enquiries welcome.

Send cheque/PO now to:-

CHEETAH MARKETING LIMITED

Dept PS 2, 24 Ray Street, London EC1 Tel: 01-278 6954

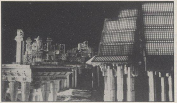
Cheetah Products also available from branches of
Rumbelows

John Menzies Boots
and WHSMITH

Peter Cameron

PLANET

Here is an object lesson in geometry.



This program began as an experiment in drawing three-dimensional shapes from different vantage points — a branch of geometry once of practical importance but now (along with much other geometric manipulations) on the decline in educational fashion. My aim was to compute and display successive views of an object as seen by a moving observer.

IN A FIX?

Let us take a fixed reference point, 0, in space as the origin, and locate any point relative to 0 by three co-ordinates (x,y,z) representing its distance, East, North (the two-dimensional map reference) and in the vertical direction. The screen should be made to display a photograph of an object taken by an observer at co-ordinates, (l,m,n). (To simplify matters, we assume that the vertical direction in space is taken to be at the top of the screen.)

Some algebraic manipulation showed that, if 0 is plotted at the centre of the screen (pixel 128,88), then the point with co-ordinates, (1,0,0) should be plotted at 128+x1, 88+y1, co-ordinates (0,1,0) at 128+x2, 88+y2, and co-ordinates (0,0,1) at 128+x3, 88+y3; it should be noted that the displacements, x1, . . .y3, are given by the formulae in lines 140 to 160 of the program. These formulae include provision for the fact that the object appears smaller when viewed from further away.

With this information, an arbitary point can be plotted — the point with co-ordinates (u,v,w) would appear at pixel, 128+u*x1+v*x2+w*x3, 88+u*y1+v*y2+w*y3.

LUCK OF THE DRAW

I decided to draw an octahedron. This figure has an interesting symmetrical shape which is extremely easy to draw. The corners, or vertices, are at the points (1,0,0), (0,1,0), (0,0,1) and their negatives. Line 260 of the program does the actual drawing.

You will see that the effect of line 260 is to draw the figure 'without taking the pencil from the paper' — that is, only using the PLOT instruction once. That this is possible for the octahedron was another reason for my choice of solid. Such a figure is called Eulerian, after the mathematician Leonhard Euler. He showed that, in order to draw a figure without lifting your pencil, it is necessary and sufficient that each vertex lies on an even number of edges (in which case, you can start anywhere but you must finish at the start point), or exactly two don't lie on the edges (when you must start at one of the edges and finish on another). It isn't necessary to demonstrate this — a routine for drawing an octahedron can be found by trial and error.

THE MECHANICS OF THE PROGRAM

The position co-ordinates of the observer are (l,m,n) and the velocity is (p,q,r). This means that, in the time interval between one observation and another, l, m and n will change by amounts proportional to p, q and r respectively. This is implemented in line 400.

At first I kept the velocity fixed. However, if the observer is piloting a spaceship orbiting a planet, the velocity will also change. According to the laws of gravity discovered by Isaac Newton, the change of velocity will be in a direction towards the planet; that is, proportional to (−l, −m, −n)/SQR b, where b is computed in line 140. The proportionality factor is the inverse square of the distance from the planet, ie 1/b.

At this point, I realised that there was a 'game' element possible within my program, so I added controls, which change the velocity either towards or away from the planet. Unfortunately, the ship's engines are not very reliable and the actual change is somewhat reliable, but that is what games are all about, aren't they? The velocity change is implemented in line 410 and 420.

WHAT'S THE SCORE

The object of the game is to take survey photographs of the planet; these should be as large as possible so you'll have to get in real close. The score is determined by the size of the largest image to appear on the screen, with added bonuses for a very close approach. However, have I told you about the risk of crashing?

Since the controls only work ▶

slowly, there is a more drastic way of avoiding an imminent crash — a jump through hyperspace. Newton's equations didn't exactly cover this possibility so I took it upon myself to assume that re-entry from hyperspace is random, both in its position and velocity. Hyperspace jumps are expensive but spectacular (especially if you re-materialise inside a solid body!)

ALL CHANGE

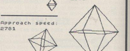
This program is written for the 16K Spectrum. However, with a few changes it could be RUN on the 48K machine; the changes are as follows:

Line 10 — replace 31743 with 64511
Line 60 — replace 31744 with 64512
Line 80 — replace 124 (the third entry in the DATA list) with 252
Line 550 — replace 31744 with 64512
Line 555 — replace 31756 with 64524
Line 600 — replace 31768 with 64536

The listing for the program. Planet.

```
  10 CLEAR 31743: BORDER 6
  20 PRINT "You are entering orb
it about"'"the octahedral planet
."'"Your aim is to make the clos
est"'"possible approach."'"If th
e image overflows the"'"screen,y
ou have crashed."'"Control may f
ail;hyperspace"'"jump will still
 operate."
  30 PRINT "Controls:"'" z for l
ower orbit"'" x for higher orbit
"'"  (each costs 1 fuel unit)"'"
 c for position and"'"  velocity
 report(free)"'" n to restore co
ntrol after"'"  failure(cost 10)
"'" m for hyperspace jump(cost 3
0)"
  40 PRINT "Your mission is comp
leted on"'"leaving the planet(im
age<5)"'"Restoring control then
costs 5"
  50 FOR i=0 TO 23
  60 READ n: POKE 31744+i,n
  70 NEXT i
  80 DATA 33,24,124,17,0,88,1,0,
3,237,176,201,33,1,88,17,0,88,1,
255,2,237,176,201
  90 GO SUB 580
 100 INPUT "Enter level(1-5)";e
 110 LET fuel=0: LET size=0: LET
 d=1: LET a$="": DEF FN f(x,y)=x
+((y-x) AND y>x): LET k=1+e/4
 120 LET p=-2*k*RND: LET q=-2*k*
RND: LET r=-2*k*RND
 130 LET l=4*RND: LET m=4*RND: L
ET n=4*RND
 140 LET a=l*l+m*m: LET b=a+n*n
 150 LET x1=40*m/(SQR a*SQR b):
LET x2=-x1*l/m: LET x3=0
 160 LET y1=40*l*n/(b*SQR a): LE
T y2=y1*m/l: LET y3=-40*SQR a/b
 170 LET s=FN f(ABS x1,ABS x2)
 180 IF s>=128 THEN GO TO 450
 190 LET w=FN f(ABS y3,ABS y2):
LET t=FN f(ABS y1,w)
 200 IF t>=88 THEN GO TO 450
 210 LET image=FN f(s,t): LET si
ze=FN f(size,image)
 220 BORDER 6: PAPER 7: FLASH 0:
 CLS
 230 IF image<5 THEN GO TO 490
 240 IF d=0 THEN PRINT AT 0,15;
INK 2; FLASH 1;"Control failure"
: GO TO 260
 245 PRINT AT 0,0; INK 0;"Approa
ch speed:";-INT (1000*(l*p+m*q+n
*r))
 250 IF a$="c" THEN PRINT INK 0;
"Coordinates:";INT (1000*l);",";
INT (1000*m);",";INT (1000*n)'"V
elocity:";INT (1000*p);",";INT (
1000*q);",";INT (1000*r)
 260 INK 1: PLOT 128+x1,88+y1: D
RAW x3-x1,y3-y1: DRAW x2-x3,y2-y
3: DRAW -x1-x2,-y1-y2: DRAW -x2+
x1,-y2+y1: DRAW -x3+x2,-y3+y2: D
RAW x1+x3,y1+y3: DRAW -x2-x1,-y2
-y1: DRAW x3+x2,y3+y2: DRAW -x1-
x3,-y1-y3: DRAW -x3+x1,-y3+y1: D
RAW x2+x3,y2+y3: DRAW x1-x2,y1-y
2
 270 INK 0: PRINT AT 21,0;"Image
:";INT image
 280 PRINT AT 21,18;"Fuel used:"
;fuel
 290 LET a$=""
 300 FOR i=1 TO 200/2↑e
 310 IF INKEY$="m" THEN GO TO 54
0
 320 IF INKEY$="" THEN GO TO 340
 330 LET a$=INKEY$
 340 NEXT i
 350 IF a$="n" AND d=0 THEN LET
d=1: LET fuel=fuel+10
 360 LET c=0
 370 IF a$="z" THEN LET c=-1
 380 IF a$="x" THEN LET c=1
 390 IF d=1 THEN LET d=1-INT (RN
D*(1+k/200))
 400 LET l=l+p/20: LET m=m+q/20:
 LET n=n+r/20
 410 LET u=b*SQR b/(2*k): LET v=
c*d/SQR b
 420 LET p=p-l/u+v*l*RND: LET q=
q-m/u+v*m*RND: LET r=r-n/u+v*n*R
ND
 430 LET fuel=fuel+ABS c*d
 440 GO TO 140
 450 REM crash
 460 BORDER 2: PAPER 2: CLS
 470 PAUSE 0
 480 GO TO 100
 490 REM mission completed
 500 CLS
 510 PRINT AT 9,7; FLASH 1;"Miss
ion completed"
 520 PRINT AT 11,12; FLASH 1;"Sc
ore:";INT (2*(size+2*((size-50)
AND size>50)+10*((size-80) AND s
ize>80)))-fuel-5*(1-d)
 530 GO TO 100
 540 REM hyperspace jump
 545 CLS
 550 LET w=USR 31744
 555 FOR i=1 TO 200: LET x=USR 3
1756: POKE 23295,128+INT (RND*64
): NEXT i
 560 LET fuel=fuel+30
 565 LET l=3*RND: LET m=3*RND: L
ET n=3*RND
 570 LET p=2*k*(2*RND-1): LET q=
2*k*(2*RND-1): LET r=2*k*(2*RND-
1)
 575 GO TO 140
 580 REM set up random attribute
 array
 590 FOR i=0 TO 767
 600 POKE 31768+i,128+INT (RND*6
4)
 610 NEXT i
 620 RETURN
```

Some sample screen illustrations from the program. Planet.



```
Approach speed:
13027
```

```
Approach speed:
-2049
```

```
Approach speed:
2781
```

'NOW WIN THE POOLS'

THIS IS MEANT FOR **YOU** — ESPECIALLY IF YOU USE **SINCLAIR SPECTRUM COMPUTER**, or even ANY COMPUTER — OR NO COMPUTER AT ALL

HAVE YOU EVER HAD THAT DREAM OR EVER WISHED THAT YOU HAD "WON ON THE POOLS" — AT LAST YOU CAN TURN IT INTO REALITY.

THERE **IS** A SECRET OF "HOW TO WIN ON THE FOOTBALL POOLS" — **IT CAN BE DONE.** I DISCOVERED THE SECRET A LONG TIME AGO — NOW, **FOR THE FIRST TIME** I'M PREPARED TO SHARE IT WITH YOU.

HOW DOES THIS INTEREST YOU — I HAVE DOCUMENTARY EVIDENCE BY WAY OF POOLS WINNINGS DIVIDEND SLIPS/CANCELLED CHEQUES, etc, SHOWING MY PRESENT WINS ON THE POOLS AS FOLLOWS:—

First Dividends	Second Dividends	Third Dividends	Fourth Dividends	Fifth Dividends	Sixth Dividends
765	1,818	2,942	1,952	631	93

A GRAND TOTAL OF 8,201 (EIGHT THOUSAND, TWO HUNDRED AND ONE DIVIDENDS — so far).

I HOLD THE UNCHALLENGED WORLD'S RECORD FOR POOLS WINS

Do not let anyone tell you that it is impossible to "WIN ON THE POOLS" — since I perfected my method, **I HAVE WON REGULARLY** for over **TWENTY-FIVE YEARS** — proof that it is no 'flash-in-the-pan'.

I have CHALLENGED THE WORLD with my record of wins and with all the evidence that I possess — NO ONE has ever been able to accept the Challenge — I KNOW NO ONE EVER WILL.

MY SECRET IS NOW PLACED ONTO COMPUTER CASSETTE FOR YOU.

THE METHOD IS THE GREATEST TREBLE CHANCE WINNER IN THE HISTORY OF FOOTBALL POOLS — IT WILL LAST FOREVER — BOTH FOR ENGLISH AND AUSTRALIAN FOOTBALL POOLS, WITH EQUAL SUCCESS.

I now intend to give a limited number of people the opportunity of making use of my method — perfected over 25 years and proving itself on **EVERY ONE OF THOSE TWENTY-FIVE YEARS.**

You will have noted details of my personal achievements so far, as given to you above.

A GRAND TOTAL of 8,201, yes 8,201 POOLS DIVIDENDS, including **765 FIRST DIVIDENDS.**

My Pools Winnings Dividend slips now number so many, that they fill a very large suitcase and will stand as my evidence of all claims in **ANY COURT OF LAW IN THE WHOLE WORLD.**

Taking just the past 25 years into consideration, I have won ON AVERAGE over 328, (THREE HUNDRED AND TWENTY—EIGHT) Pools Dividends **EVERY YEAR** — or — AN AVERAGE of over **SIX DIVIDENDS EVERY WEEK** for **TWENTY—FIVE YEARS.**

You have my absolute Guarantee of the complete authenticity of every claim, cheque, document, letter, etc, contained herein.

I do have losing weeks, but ON AVERAGE my winnings show over SIX DIVIDENDS **EVERY WEEK** for the past 25 years.

I know that you are now utterly flabbergasted, it always happens to everyone with whom I come into contact. Please just sit back and **imagine** for a moment my **FIRST DIVIDEND** wins alone — they now number 765 (seven hundred and sixty-five) and will probably be even more by the time this advertisement appears in print.

I AM NUMBER ONE IN THE WORLD AND NO ONE DISPUTES IT.

For as long as I continue to enter the Football Pools my wins will continue. I have already said, they apply, with equal success to both English and Australian Football Seasons.

I intend to release a STRICTLY LIMITED NUMBER of copies of my cassette, — DO NOT DELAY AND FIND YOU ARE TOO LATE, in which case I would have to refund your money.

I am so confident of **YOUR** success that if do **not** win at least THREE FIRST TREBLE CHANCE DIVIDENDS in the first 20 weeks of entering, I will completely cancel the balance of the purchase price and you do not have to pay me another penny, at any time, no matter how vast your winnings.

I only wish that space would allow me to give you photographs of my winnings slips, cancelled cheques, etc, but it is of course impossible — they now number 8,201 dividends. I have however given JUST A FEW EXTRACTS from ORIGINAL LETTERS I hold from my small Clientele.

I am the Inventor and Sole Proprietor of my method, Registered as EUREKA — ('I have found it'). I am known as The Professor in Pools Circles — I am of the Highest Rank in Forecasting — this is beyond dispute. I am marketing a limited number of Computer Cassettes, under my Registered Company — FOOTBALL ENTERPRISES.

My initial charge for a copy was £75, but for this SPECIAL REDUCED PRICE OFFER I will send you a copy, for £20, (twenty pounds) ONLY, plus your Promise to pay me the balance of £55 — ONLY IF YOU WIN AT LEAST THREE FIRST TREBLE CHANCE DIVIDENDS IN YOUR FIRST 20 WEEKS OF ENTERING — otherwise you owe me NOTHING FURTHER.

This is surely proof absolute of my supreme and utter confidence in my own abilities and in the capabilities of my discovery. I could easily CHARGE £2,000 per cassette on the evidence I possess, but that would not be fair to everyone, which is what I want to do.

My method is WORLD COPYWRIGHT, any infringement and immediate proceedings will be taken, without prior warning. It is truly ingenious and has stood the test of time.

My cassette is simplicity itself to operate and you'll be given FULL DETAILS for weekly calculating. Your entry need not involve you in any large weekly stakes, you can enter for as little as 25p, if you wish.

I charge NO COMMISSION on any of your wins — no matter how BIG they may be.

I realised a long time ago, that it was no good sitting down and **dreaming** about winning the pools, so I burnt the candle at both ends, working late into the night, occasionally RIGHT THROUGH THE NIGHT, I **KNEW** there was a way, eventually it all paid off and has been doing so ever since.

I am unable to vary my offer to anyone, so please do not request it, as I shall very easily dispose of the cassettes I have prepared and am making available.

IMMEDIATELY I perfected my method I commenced winning right away, (first with just a little £163, the first week I used it), I HAVE NEVER LOOKED BACK SINCE, amongst all those dividends was one for over **EIGHT THOUSAND POUNDS** for just **one eighth of a penny stake.**

I will release a copy on cassette, to you, on receipt of the completed order form and your Signature thereon, confirming you will treat it in the STRICTEST CONFIDENCE between us and retain it for your **OWN USE ONLY.**

PLEASE NOTE:
If you happen to be the proud owner of a Computer **other than** Sinclair Spectrum, you can still purchase a copy of my method, for the same price and program it YOURSELF on to YOUR OWN COMPUTER — or even if you do not have a computer.

Don't take my word for it, read what people write about me and my method:—

I won on Zetters last weekend. It was not a big sum, but all the same it was a very nice surprise for me. J.C., Lancs.

I appreciate the straightforward method you adopt, which is such a contrast to the rubbish of misrepresentation which is so common in the Betting World, by unscupulous and self-opinionated charlatans. C.H., Devon

Winnings cheque received today, sincere thanks. D.N., Devon

I congratulate you on your achievement. R.R., Wales

I should like to thank you for a most exciting season and look forward to hearing from you again. J.C., Hants.

I would like to acknowledge cheque and say how much I appreciate your integrity. J.M., Scotland

Many thanks for your system, it is all you say and more. J.C., Lancs.

Your wonderful system won me £3,527. I intend to visit London soon and will be able to come and see you personally. (Overseas Client). P.M., Kampala.

Many thanks for trying so hard to please us all, your brother should be thanked also. One of our daughters, WHOSE HUSBAND YOU HELPED ENORMOUSLY, has just phoned, the four of them have just spent a lovely holiday in Spain. K.R.,Isle of Man.

I sent in my FIRST entry last week and won 2nd and 3rd dividends, as you will see from the enclosed certificate. One more and I would have collected over £400 for FIRST dividend. Once I've won a fair amount I shall be staking from winnings and at 2p per line, A FIRST DIVIDEND last week at this would have been over £3,000. C.A., Yorks.

I am very interested indeed and enclose £20 herewith. I agree to pay you the balance of £55 **ONLY** if I win at least THREE FIRST TREBLE CHANCE DIVIDENDS in my first 20 weeks of entering — otherwise I owe you NOTHING FURTHER at any time — no matter how much money I win. My Signature below is my Undertaking to retain complete and absolute confidence about the method.

Name……………………………………………………………

Address…………………………………………………………

…………………………………………………………………

…………………………………………………………………

Signature………………………………………………………

The Managing Director,
Football Enterprises,
'Anvon',
9 New Road,
Haverfordwest, Pembs.

Please tick if cassette is for:
Sinclair Spectrum (48K) ☐
Any other Computer ☐
No Computer at all ☐

C Taft

PLAY YOUR CARDS RIGHT

See if you can make this pontoon game stick on your Spectrum.

This program simulates the game of pontoon, in which you get to pit your wits against the computer.

When RUN, you will be dealt a card and asked to place a bet; bets can only be made up to a certain amount – within 10% of your credit. You will then be dealt another card and given the option to 'stick', 'twist' or 'buy'. This is done by pressing the appropriate key on the keyboard ie the 's' key to stick, the 't' key to twist and the 'b' key to buy.

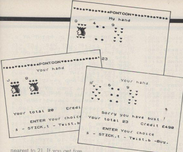
YOUR DEAL . . .

If you've never played pontoon before, here is a brief resumé. You are dealt a single card, and on that you have to work out how good your hand might be and place a suitable bet accordingly. You then receive your next card. The idea of the game is to make your cards up to a value of 21; King, Queens and Jacks are all worth 10 points each, and an Ace can be worth one or 11. With the two original cards you hold, you must decide either to 'stick' which means you pick up no more cards, 'twist' which means you receive another card (you can receive up to three more cards), or 'buy' which means you receive another card but you buy it from the dealer so that you effectively increase your bet.

The winner of a hand of pontoon is the one who gets their card to total 21 or the

AMT – Amount of credit.
ITME – The Spectrum or player dealing.
ACEI – An ace has been dealt.
ME – Value of the player's final hand.
V – Current value of the cards.
T – Number of cards dealt.
ODDS – Calculation for the Spectrum to decide whether to stick or not.
Bet – Amount bet on the initial card.
Buy – Total amount for bought cards.
Stake – The stake for each game.
S$ – The suit of each card.
R$ – The Jack, Queen or King.

The variables used in this program.



```
♥♠♦♥♠♦♥♠♦PONTOON♥♠♦♥♠♦♥♠♦
            My hand
Q      4       9
```

```
♥♠♦♥♠♦♥♠♦PONTOON♥♠♦♥♠♦♥♠♦
        Your hand
J     Q

Your total 20       Credi

     ENTER Your choice

 s - STICK,t - Twist,b
```

```
          Your hand
8      6       9

   Sorry you have bust !
Your total 23       Credit £490
       ENTER Your choice
 s - STICK,t - Twist.b -Buy.
```

nearest to 21. If you get five cards to total 21 or under, then this is known as a 'five card trick' and this hand beats all. Should the computer and your hand equal the same, it will depend on who is dealing as to who will win — in the game of pontoon, this is known as Banker's privilege.

TWIST AND SHOUT

When you have settled on your final hand and pressed the 's' key to stick, then the computer deals its own hand, decides whether to stick or not and finally checks its hand against your own. The computer then either takes your bet if it has won or returns your original bet plus your winnings if you managed to have the winning hand.

One nice touch in the program is that the computer doesn't just keep dealing itself cards until it beats your own hand or 'goes bust' (which is pontoon-talk for holding cards which total more than 21) but actually makes a decision whether or not to draw another card. This decision is based on the value of the cards you have stuck on and the amount of cash you have placed as your bet. Only when the computer has stuck are the cards displayed and the winner announced.

VARIATIONS ON A THEME

Although there are copious REM statements throughout the program to help you follow its intricacies, a list of the variables used is given.

```
  20 POKE 23609,30: GO TO 1850
  30 LET amt=500
  40 LET stake=10: DIM z$(1,8):
DIM s$(4): DIM c(52): DIM h(5,2)
: DIM t(4): LET s$(1)="H": LET s
$(2)="D": LET s$(3)="S": LET s$(
4)="C"
  50 LET itme=100: LET ace1=0: L
ET w=0: LET k=1: LET a=1: LET c=
0: LET me=0: LET v=0: LET buy=0:
 LET bet=0: LET amt=amt-stake: L
ET n=0
  60 FOR s=1 TO 4
  70 FOR r=2 TO 14
  80 LET n=n+1: LET c(n)=10*r+s
  90 NEXT r: NEXT s
 100 FOR n=1 TO 52
 110 BEEP .003,-n+52: LET m=INT
(RND*52)+1: LET t=c(m): LET c(m)
=c(n): LET c(n)=t
 120 BEEP .003,n: NEXT n
 130 LET n=0
 140 PAPER 4: BORDER 2: CLS
 150 FOR n=0 TO 8 STEP 4: PAPER
7: INK 2: PRINT AT 0,0+n;"H"; IN
K 0;"C"; INK 2;"D"; INK 0;"S";:
NEXT n
 160 PAPER 1: INK 7: PRINT "PONT
OON";
 170 FOR n=19 TO 27 STEP 4: PAPE
R 7: INK 2: PRINT AT 0,n;"H"; IN
K 0;"C"; INK 2;"D"; INK 0;"S";:
NEXT n: PRINT INK 2;"H"
 180 PAPER 7: INK 1: PRINT AT 2,
11;"Your hand";AT 15,2;"Your tot
al ";v;AT 15,19;"Credit £";amt
 190 GO SUB 580
 200 IF ace1=0 AND t>=2 THEN GO
TO 310
 210 GO TO 510: GO TO 470
 220 REM S T B
 230 PAPER 4: PRINT AT 18,0;"
     "; PAPER 7; INK 2;AT 18,7;"E
NTER "; INK 1;"Your choice"; PAP
ER 4;"
                   "; PAPER 7;"s -
STICK,t - Twist.b -Buy."
 240 IF t=5 THEN LET me=21: GO T
O 1550
 250 INPUT a$
 260 IF a$<>"s" AND a$<>"t" AND
a$<>"b" THEN GO TO 250
 270 IF a$="s" THEN PAUSE 50: GO
 TO 470
 280 IF a$="b" THEN GO TO 340
 290 REM T
 300 LET t=t+1: GO SUB 600
 310 REM I BUST
 320 IF v>21 THEN GO TO 410
 330 GO TO 220
 340 REM I BUY
 350 PAPER 7: INK 2: PRINT AT 18
,2;"ENTER"; INK 1;" Amount (Whol
e £'s)."
 360 PAPER 7: INK 2: PRINT AT 20
,2;"UPTO "; INK 2;" £";INT (((am
t-bet-buy)*.1)+.5); PAPER 4;"
                      "

 370 INPUT buy1
 380 IF buy1>INT ((amt-bet-buy)*
.1)+.5 THEN GO TO 370
 390 LET buy=buy+buy1
 400 PAUSE 50: GO TO 290
 410 REM I BUST
 420 PAUSE 20: FOR z=1 TO 40: BE
EP .05/(z/7),-z/7: FLASH 1: PRIN
T AT 13,6; PAPER 7; INK 2;"Sorry
 you have bust": NEXT z: FLASH 0
 430 PAPER 4: PRINT AT 13,6;"
                      "
 440 LET amt=INT (amt-(buy+bet)+
.5)
 450 PAPER 7: INK 1: PRINT AT 15
```

```
,27;amt; PAPER 4;" "
 460 GO TO 50
 470 REM S
 480 IF v>=15 THEN LET me=v: GO
TO 860
 490 BEEP .5,-5: PAPER 3: INK 7:
PRINT AT 13,0;"You cannot stic
on less than 15": PAUSE 75
 500 PAPER 4: PRINT AT 13,0;"
                              ":
GO TO 220
 510 REM Bet
 520 PAPER 7: INK 2: PRINT AT 18
,2;"ENTER"; INK 1;" Your bet (Wh
ole £'s).";AT 20,2; INK 2;"UPTO
"; INK 1;" £";INT (amt*.1)+.5;
 530 PAPER 4: PRINT "
      "
 540 INPUT bet
 550 IF bet>INT (amt*.1)+.5 THEN
GO TO 540
 560 IF ace1=1 THEN GO SUB 600:
GO TO 1720
 570 LET t=t+1: GO SUB 600: GO T
O 220
 580 REM PICK CARD
 590 LET t=1
 600 LET d=1: LET n=n+1: LET h(t
,d)=c(n)*10+t: LET temp=c(n): GO
SUB 760
 610 PAUSE 70
 620 GO SUB 1780
 630 IF r$(1)="j" OR r$(1)="q" O
R r$(1)="k" THEN BEEP .1,r: GO T
O 650
 640 BEEP .1,r: GO SUB 1270+(r*2
0)
 650 IF r$(1)="j" THEN GO SUB 14
90
 660 IF r$(1)="q" THEN GO SUB 15
10
 670 IF r$(1)="k" THEN GO SUB 15
30
 680 IF itme=100 THEN INK 1: PRI
NT AT 15,13;v
 690 IF itme<>100 THEN PAPER 7:
INK 1: PRINT AT 15,2;"My total "
;v;AT 15,19;"pay ";AT 19,2;"Amou
nt bet £";bet+buy;AT 19,19;"Cred
it £";amt
 700 IF itme<>100 AND v>=21 OR i
tme<>100 AND t=5 THEN GO TO 740
 710 IF itme<>100 AND v=20 THEN
PRINT v+1;AT 17,14;"Five card tr
icks"
 720 IF itme<>100 THEN PRINT AT
15,23;v+1
 730 IF itme<>100 THEN GO TO 750
 740 IF itme<>100 THEN PAPER 4:
PRINT AT 15,19;"      "
 750 RETURN
 760 LET r=INT (temp/10): LET s=
temp-r*10: LET r=r+1: LET r$=STR
$ r
 770 IF r=15 THEN LET r$="ACE"
 780 IF r=15 AND ace1=0 THEN GO
TO 1680
 790 IF r=15 AND ace1=1 THEN GO
SUB 600
 800 IF r=14 THEN LET r$(1)="k"
 810 IF r=13 THEN LET r$(1)="q"
 820 IF r=12 THEN LET r$(1)="j"
 830 IF r>=12 AND r<15 THEN LET
r=11
 840 LET r=r-1: LET v=v+r
 850 RETURN

 860 REM ZX deal
 870 CLS : PAPER 4
 880 FOR z=0 TO 8 STEP 4: PAPER
7: INK 2: PRINT AT 0,0+z;"H"; IN
K 0;"C"; INK 2;"D"; INK 0;"S";:
NEXT z
 890 PAPER 1: INK 7: PRINT "PONT
OON";
 900 FOR z=19 TO 27 STEP 4: PAPE
R 7: INK 2: PRINT AT 0,z;"H"; IN
K 0;"C"; INK 2;"D"; INK 0;"S";:
NEXT z
```

```
 910 PAPER 7: INK 2: PRINT AT 2,
13;"My hand"
 920 LET c=c+6: LET v=0: LET v1=
0: LET itme=0: LET t=0: LET a=1
 930 IF t=5 AND v<=21 THEN GO TO
1030
 940 IF v<15 THEN GO TO 1000
 950 IF v=21 THEN GO TO 1030
 960 IF v>21 THEN GO TO 1100
 970 LET odds=INT ((v/((bet+buy)
(amt/5)))/50)+1
 980 IF odds<4 AND v>=17 THEN GO
TO 1010
 990 IF odds>=4 THEN GO TO 1010
1000 LET t=t+1: GO SUB 600: GO T
O 930
1010 IF v>=me OR v=me+1 THEN GO
TO 1030
1020 IF v<me+1 THEN GO TO 1100
1030 REM I lose
1040 LET amt=INT (amt-(bet+buy)+
.5)
1050 PAPER 3: INK 7: PRINT AT 19
,2;"You've lost £";bet+buy
1060 PAPER 7: INK 1: PRINT AT 19
,27;amt; PAPER 4;"    "
1070 FOR x=20 TO -10 STEP -.7: B
EEP .01,x: NEXT x
1080 IF amt<20 THEN PAPER 7: INK
 2: FOR z=100 TO 50 STEP -1: BEE
P .01,z-75: FLASH 1: PRINT AT 13
,4;"Sorry no money left!!!!": NE
XT z: FLASH 0: GO TO 1220
1090 GO TO 50
1100 REM I WIN
1110 LET wl=bet+buy+stake
1120 LET amt=amt+wl
1130 LET amt=INT (amt+.5)
1140 PAPER 2: INK 7: PRINT AT 19
,2;"You've won £";bet+buy;
1150 PAPER 4: PRINT AT 15,19;"
   "
1160 IF v=20 THEN PAPER 4: PRINT
 AT 17,14;"
                             "
1170 PAPER 7: INK 1: PRINT AT 19
,27;amt; PAPER 4;" "
1180 FOR z=1 TO 4: BEEP .2,0: PA
USE 4: BEEP .2,-5: PAUSE 4: BEEP
 .2,5: PAUSE 4: NEXT z
1190 REM HIGH AMT
1200 IF amt<999 THEN GO TO 50
1210 PAPER 7: INK 2: FOR z=50 TO
 100: FLASH 1: PRINT AT 13,3;"Yo
u have broke the bank!!!": BEEP
.01,z-75: NEXT z: FLASH 0
1220 PAPER 4: PRINT AT 13,2;"
     "; PAPER 7; INK 2;"Another
game?"; PAPER 4;"           "
1230 INPUT a$
1240 IF a$="y" THEN GO TO 30
1250 IF a$="n" THEN STOP
1260 IF a$<>"y" OR a$<>"n" THEN
GO TO 1230
1270 REM Print cards
1280 REM 1
1290 GO SUB 1780
1300 PAPER 7: PRINT AT k+3,a;"A
    ";AT k+4,a;"      ";AT k+5,a;
     ";AT k+6,a;"  ";CHR$ z;"
;AT k+7,a;"      ";AT k+8,a;"
  ";AT k+9,a;"      ": LET a=a+6:
RETURN

1310 REM 2
1320 PAPER 7: PRINT AT k+3,a;"2
    ";AT k+4,a;"  ";CHR$ z;"  ";A
T k+5,a;"      ";AT k+6,a;"
;AT k+7,a;"      ";AT k+8,a;"  ";
CHR$ z;"  ";AT k+9,a;"      ": LE
T a=a+6: RETURN
1330 REM 3
1340 PAPER 7: PRINT AT k+3,a;"3
    ";AT k+4,a;"  ";CHR$ z;"  ";A
T k+5,a;"      ";AT k+6,a;"  ";CH
R$ z;"  ";AT k+7,a;"      ";AT k+
8,a;"  ";CHR$ z;"  ";AT k+9,a;"
    ": LET a=a+6: RETURN
1350 REM 4
1360 PAPER 7: PRINT AT k+3,a;"4
```

```
  ";AT k+4,a;" ";CHR$ z;" ";CHR
$ z;" ";AT k+5,a;"     ";AT k+6,
a;"     ";AT k+7,a;"     ";AT k+
8,a;" ";CHR$ z;" ";CHR$ z;" ";AT
 k+9,a;"     ": LET a=a+6: RETUR
N
1370 REM 5
1380 PAPER 7: PRINT AT k+3,a;"5
  ";AT k+4,a;" ";CHR$ z;" ";CHR
$ z;" ";AT k+5,a;"     ";AT k+6,
a;" ";CHR$ z;" ";AT k+7,a;"
  ";AT k+8,a;" ";CHR$ z;" ";CHR$
 z;" ";AT k+9,a;"     ": LET a=a
+6: RETURN
1390 REM 6
1400 PAPER 7: PRINT AT k+3,a;"6
  ";AT k+4,a;" ";CHR$ z;" ";CHR
$ z;" ";AT k+5,a;"     ";AT k+6,
a;" ";CHR$ z;" ";CHR$ z;" ";AT k
+7,a;"     ";AT k+8,a;" ";CHR$ z
;" ";CHR$ z;" ";AT k+9,a;"
: LET a=a+6: RETURN
1410 REM 7
1420 PAPER 7: PRINT AT k+3,a;"7
  ";AT k+4,a;" ";CHR$ z;" ";CHR
$ z;" ";AT k+5,a;"  ";CHR$ z;"
";AT k+6,a;" ";CHR$ z;" ";CHR$ z
;" ";AT k+7,a;"     ";AT k+8,a;"
 ";CHR$ z;" ";CHR$ z;" ";AT k+9,
a;"     ": LET a=a+6: RETURN
1430 REM 8
1440 PAPER 7: PRINT AT k+3,a;"8
  ";AT k+4,a;" ";CHR$ z;" ";CHR
$ z;" ";AT k+5,a;" ";CHR$ z;"
";AT k+6,a;" ";CHR$ z;" ";CHR$ z
;" ";AT k+7,a;" ";CHR$ z;"  ";A
T k+8,a;" ";CHR$ z;" ";CHR$ z;"
";AT k+9,a;"     ": LET a=a+6: R
ETURN
1450 REM 9
1460 PAPER 7: PRINT AT k+3,a;"9
  ";AT k+4,a;" ";CHR$ z;" ";CHR
$ z;" ";AT k+5,a;" ";CHR$ z;" ";
CHR$ z;" ";AT k+6,a;"  ";CHR$ z;
"  ";AT k+7,a;" ";CHR$ z;" ";CHR
$ z;" ";AT k+8,a;" ";CHR$ z;" ";
CHR$ z;" ";AT k+9,a;"     ": LET
 a=a+6: RETURN
1470 REM 10
1480 PAPER 7: PRINT AT k+3,a;"T
  ";AT k+4,a;" ";CHR$ z;CHR$ z;
CHR$ z;" ";AT k+5,a;" ";CHR$ z;
"  ";AT k+6,a;" ";CHR$ z;" ";CHR
$ z;" ";AT k+7,a;"  ";CHR$ z;"
";AT k+8,a;" ";CHR$ z;CHR$ z;CHR
$ z;" ";AT k+9,a;"     ": LET a=
a+6: RETURN
1490 REM J
1500 PAPER 7: PRINT AT k+3,a;"J
  ";AT k+4,a;" ";CHR$ z;CHR$ z;
CHR$ z;" ";AT k+5,a;" ABE ";AT k
+6,a;" IJK ";AT k+7,a;" LMN ";AT
 k+8,a;" ";CHR$ z;CHR$ z;CHR$ z;
" ";AT k+9,a;"     ": LET a=a+6:
 RETURN
1510 REM Q
1520 PAPER 7: PRINT AT k+3,a;"Q
  ";AT k+4,a;" ";CHR$ z;CHR$ z;
CHR$ z;" ";AT k+5,a;" FGO ";AT k
+6,a;" IJK ";AT k+7,a;" LMN ";AT
 k+8,a;" ";CHR$ z;CHR$ z;CHR$ z;
" ";AT k+9,a;"     ": LET a=a+6:
 RETURN
1530 REM K
1540 PAPER 7: PRINT AT k+3,a;"K
  ";AT k+4,a;" ";CHR$ z;CHR$ z;
CHR$ z;" ";AT k+5,a;" PQR ";AT k
+6,a;" IJK ";AT k+7,a;" LMN ";AT
 k+8,a;" ";CHR$ z;CHR$ z;CHR$ z;
" ";AT k+9,a;"     ": LET a=a+6:
 RETURN
1550 REM SCT
1560 PAPER 7: INK 2: FOR z=50 TO
 100: FLASH 1: BEEP 1/z,z-75: PR
INT AT 13,7;"Five card trick": N
EXT z: FLASH 0
1570 PAPER 4: PRINT AT 13,7;"
               ": GO TO 860
1580 REM ZX 11/1 ?
```

```
1590 PAUSE 70
1600 IF t=1 THEN LET r=11: GO TO
 1650
1610 IF t=4 OR v<7 THEN LET r=1:
 GO TO 1650
1620 IF t=2 AND v<=10 AND v>=7 O
R t=3 AND v<=10 AND v>=7 OR t=4
AND v<=10 AND v>=7 THEN LET r=11
: GO TO 1650
1640 LET r=1
1650 LET v=v+r
1660 GO SUB 1780
1670 BEEP .1,r: GO SUB 1290: GO
SUB 680: GO TO 930
1680 REM ZX/ME?
1690 REM ME ACE 1/11
1700 IF itme=0 THEN GO TO 1580
1710 IF r$(1)="A" AND t=1 THEN L
ET r=0: GO SUB 1290: LET ace1=1:
 GO TO 510
1720 PAPER 7: INK 2: PRINT AT 16
,0;"Ace dealt do you want 1 or 1
1?"; PAPER 4;"
              "
1730 INPUT r
1740 IF ace1=1 THEN LET v=v+r: G
O SUB 680
1750 IF ace1=1 THEN LET ace1=0:
GO TO 220
1760 IF ace1=0 THEN BEEP .1,r: G
O SUB 1290
1770 LET v=v+r: GO SUB 680: GO T
O 200
1780 REM SUIT COLOUR
1790 IF s$(s)="H" THEN LET z=151
: INK 2
1800 IF s$(s)="C" THEN LET z=146
: INK 0
1810 IF s$(s)="D" THEN LET z=147
: INK 2
1820 IF s$(s)="S" THEN LET z=162
: INK 0
1830 RETURN
1840 REM GRAPHICS
1850 RESTORE : FOR n=1 TO 20: RE
AD p$
1860 FOR f=0 TO 7
1870 READ a: POKE USR p$+f,a
1880 NEXT f
1890 NEXT n
1900 DATA "h",0,0,54,62,28,8,0,0
1910 DATA "d",0,0,8,28,62,28,8,0
1920 DATA "c",0,0,28,42,62,42,8,
8
1930 DATA "s",0,0,8,28,62,54,8,0
1940 DATA "t",0,94,82,82,82,82,9
4,0
1950 DATA "a",0,0,243,121,63,31,
15,7
1960 DATA "b",0,0,231,231,255,25
5,255,255
1970 DATA "e",0,0,207,158,252,24
8,240,244
1980 DATA "i",4,12,14,16,32,64,6
4,56
1990 DATA "j",84,85,42,21,10,10,
10,5
2000 DATA "k",160,80,144,88,164,
172,162,86
2010 DATA "l",8,4,24,16,16,15,1,
3
2020 DATA "m",5,5,5,29,98,191,0,
0
2030 DATA "n",82,86,82,82,212,88
,232,4
2040 DATA "f",0,0,240,121,63,31,
15,7
2050 DATA "g",0,0,195,231,255,60
,255,255
2060 DATA "o",0,0,15,158,252,248
,240,224
2070 DATA "p",0,0,255,121,48,16,
8,7
2080 DATA "q",0,0,255,231,195,19
5,231,25
2090 DATA "r",0,0,255,158,12,8,1
44,224
2100 GO TO 30
2110 SAVE "pontoon" LINE 1
```

Chris Lemon

DEMOLITION

Get rid of your aggressive tendencies and have a smashing time on your Spectrum

In this game for your 16K Spectrum, the player is presented with a series of brick walls which slowly move up the screen. The walls must be demolished by dropping an explosive asterisk down onto sections of the walls. The game is over when the wall reaches the top of the screen and the asterisk bumps into it.

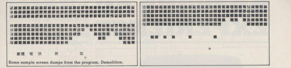
The program has two features:

- The score is printed in the input area.
- The walls are advanced by scrolling with SCR CT supressed.

The directions for play are very simple - once typed in and RUN, you simply press Enter to start the game and press the '6' key to drop the explosive asterisk as it moves across the top of the screen. The asterisk can move from right to left or *vice versa;* there is no warning given as to which side the asterisk will come.

Demolition is a simple, though very addictive game.

Photo courtesy of Cinema International Corp. (UK).



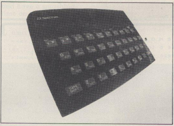
Some sample screen dumps from the program. Demolition.

```
  40 RANDOMIZE
  50 BORDER 6
  60 PRINT AT 10,6;"D E M O L I
T I O N"
  70 PRINT AT 15,4;"Use 6 to bre
ak the walls."
  80 PRINT AT 16,4;"Press enter
to start."
  90 INPUT LINE l$
  95 CLS
 100 LET c=0: LET k=8: LET a=0:
LET s=0: LET p=1
 190 GO TO 8000
 200 REM
 202 GO SUB 4000
 210 FOR i=0 TO 31
 211 IF SCREEN$ (1,i)="" THEN GO
 TO 3000
 213 NEXT i
 220 LET c=c+1: IF c=k THEN GO T
O 1000
 240 LET vv=INT (RND*2)
 250 IF vv=0 THEN GO TO 300
 260 GO TO 350
 300 REM go right
 310 PRINT AT 1,0;"*";
 312 IF INKEY$="6" THEN GO TO 50
0
 320 FOR i=1 TO 31
 322 BEEP .01,16: PAUSE p
 330 PRINT CHR$ 8;" ";" ";CHR$ 8
;"*";
 332 IF INKEY$="6" THEN GO TO 50
0
 340 NEXT i
 342 PRINT CHR$ 8;" ";
 348 GO TO 200
 350 REM go left
 360 PRINT AT 1,31;"*";
 362 IF INKEY$="6" THEN GO TO 50
0
 370 FOR i=0 TO 30
 372 BEEP .01,16: PAUSE p
 380 PRINT CHR$ 8;" ";CHR$ 8;CHR
$ 8;"*";
 382 IF INKEY$="6" THEN GO TO 50
0
 390 NEXT i
 392 PRINT CHR$ 8;" ";
 398 GO TO 200
 500 REM drop
 510 LET x=33-PEEK 23688-1
 520 FOR i=2 TO 20
 530 PRINT AT i-1,x;" ";
 540 IF SCREEN$ (i,x)="" THEN GO
 TO 700
 542 PRINT AT i,x;"*";: BEEP .01
,8
 550 NEXT i
 552 PRINT AT 20,x;" ";
 560 GO TO 200
 700 REM percu
 710 PRINT AT i,x;" ": LET s=s+1
0: BEEP .02,16
 712 LET i=i+1: IF i>21 THEN GO
TO 720
 714 IF SCREEN$ (i,x)="" THEN LE
T s=s+10
 716 PRINT AT i,x;" "
 720 FOR j=1 TO 5
 722 LET xm=x-j: IF xm<0 THEN LE
T xm=0
 724 LET xp=x+j: IF xp>31 THEN L
ET xp=31
 726 LET sm=(SCREEN$ (i-j,xm)=""
)
 727 LET tm=(SCREEN$ (i-j-1,xm)=
"")
 728 LET sp=(SCREEN$ (i-j,xp)=""
)
 729 LET tp=(SCREEN$ (i-j-1,xp)=
"")
 730 LET w=sm+sp+tm+tp
 740 IF w=0 THEN GO TO 200
 760 PRINT AT i-j,xm;" ";: PRINT
 AT i-j,xp;" ";
 770 PRINT AT i-j-1,xm;" ";: PRI
NT AT i-j-1,xp;" ";
 772 LET s=s+w*10: BEEP .05,4
 780 NEXT j
 810 GO TO 200
1000 LET c=0: LET a=a+1: IF a>5
THEN GO TO 1020
1010 POKE 23692,255: PRINT AT 21
,31;b$: PRINT AT 20,0;b$: GO TO
1100
1020 LET q=INT (RND*7)
1022 POKE 23692,255: PRINT AT 21
,31;b$: PRINT INK q;AT 20,0;a$:
IF a=10 THEN LET a=0
1030 GO TO 1100
1100 LET k=k-1: IF k=2 THEN LET
k=3
1110 GO TO 200
2100 LET a$="GGGGGGGGGGGGGGGGGG
GGGGGGGGGGGGGG"
2110 LET b$="                  "
2112 LET z$="     .....AAAARGHHH
H....."
2120 FOR i=0 TO 15
2130 PRINT b$
2140 NEXT i
2150 FOR i=16 TO 20
2160 PRINT INK (i-16);a$
2170 NEXT i
2180 PRINT b$
2190 GO TO 200
3000 REM stop
3010 PRINT AT 21,0;z$
3020 PAUSE 0
3024 PRINT AT 0,0;s
3030 STOP
4000 REM score
4020 LET s$=STR$ s: LET l=LEN s$
4030 FOR j=1 TO l
4040 LET f=CODE s$(j)
4050 FOR i=0 TO 7: LET e=PEEK (u
+f*8+i): POKE (v+j-1+256*i),e: N
EXT i
4060 NEXT j
4100 RETURN
8000 LET g0=BIN 00000000
8010 LET g1=BIN 01111110
8020 POKE USR "g"+0,g0
8030 FOR i=1 TO 6
8040 POKE USR "g"+i,g1
8050 NEXT i
8060 POKE USR "g"+7,g0
8080 LET u=PEEK (23607)*256+PEEK
 (23606)
8090 LET v=16*256+32*6+16384
8100 GO TO 2100
```

Peter Shaw

DRAWING ON EXPERIENCE

We hold tutorial on the subject of Spectrum graphics. Pull up a chair, plug in your computer and read on...



There are a number of different methods to define characters on the Spectrum — some easy to understand, others totally incomprehensible. I saw one very interesting method POKEd, line by line. This is obviously not satisfactory when you have 21 characters to define. You would need 168 lines to POKE everything into the memory — that's lines 10 to 1680 using the conventional spacing system! There has to be a better method than that.

To cut down on the amount of typing we have to do, we can change the way we define our characters. To start with I will deal with the use of BIN. The Sinclair manual suggests that you use BIN when defining but I think this is a very long-winded way of doing it.

A binary number can be thought of as eight on/off switches each with a value; when the switch is on, it takes the value and when the switch is off, it becomes zero. The value of each switch is worked out from a power, ie 6 2 is six to the power two, or 36.

For example, take a look at the binary number 00110101. The value of BIN 00110101 is 0 + 0 + 32 + 16 + 0 + 4 + 0 + 1 = 53. This can be seen fully illustrated in Fig. 1.

AND NEXT...

To turn a binary number into decimal all you need remember is the number in the top left-hand corner of the boxes so that you can add the 'on' ones together to get the result. Having converted all your BIN numbers into decimal you can now move onto the next part of this character generation method.

The Sinclair manual suggests you use a FOR...NEXT loop of zero to seven for some characters. I also use a FOR...NEXT loop, but I use it to cover all the characters to be defined, ie FOR A=USR "a" TO USR "(last character to be defined)" +7. Inside the loop I READ all the decimal values and POKE them into the variable 'a'.

```
10  FOR A=USR "a"
    TO USR "a"+7:
    REM this will
    require eight
    numbers in DATA
    statements
20  READ user: POKE
    a, user
30  NEXT A
40  DATA
    255,0,255,0,255-
    ,0,255,0
```

Try this program above. As you can see, the FOR...NEXT loop can be anything from one to 21 characters long. The DATA statement at the end contains the eight numbers required to create a striped character.

AT THE BAR

Enough said about user-definable graphics — what about programs which demonstrate high resolution graphics on the Spectrum? Well, here's a 3D Histogram program, which although not really very useful in any practical way, is a good demonstration of colour and Hi-res in action. The program surprisingly only uses two UGD characters, which have been POKEd in using the method shown previously.

QUICK ON THE DRAW

Once you've tried the Histogram program, you can move onto the next program, Sketch, which is a very sophisticated sketchpad.

The controls for movements are shown in Fig. 2. The other controls you'll need to operate this program are:

F — Flashing cursor.
N — Draw mode cursor.
D — Draw a line.
M
— Move cursor.
S — Save screen$.
V — CLS.
C — Circle.
O — Change INK colour.
P — Change PAPER colour.
B — Change BORDER colour.
1 — Help.

The 3D Histogram program.

```
  30 GO SUB 9000
  40 GO SUB 6000
  50 GO SUB 7000
  60 LET d=1
  70 FOR a=10 TO 0 STEP -1
  80 PRINT AT d,0;a
  90 LET d=d+2
 100 NEXT a
 130 INPUT "How many Inputs (Max
 25) ";i: IF i>25 THEN GO TO 130
 135 IF i<14 THEN LET p=2
 136 IF i>=14 THEN LET p=1
 140 FOR a=4-(1 AND p=1) TO ((i-
1)*p)+4-(1 AND p=1) STEP p
 150 INPUT "Input ";(INT (a-2)/p
);h
 155 IF h>10 THEN GO TO 150
 160 LET h1=(h*2)-1
 165 PRINT AT 21,a; INK 0; PAPER
 4;"A"; PAPER 2;"B"
 170 FOR b=20 TO 21-(h1+1) STEP
-1
 180 PRINT AT b,a; INK 4; PAPER
6;"A"; INK 2;"B"
 190 PRINT AT b,a; PAPER 4;" ";
PAPER 2;" "
 200 NEXT b
 210 PRINT AT b,a; PAPER 8; INK
6;"BA"
 220 LET b=b+1
 230 PRINT AT b,a; INK 4; PAPER
6;"A"; INK 2;"B"
 240 NEXT a
 250 INPUT "Press ENTER for anot
her run "; LINE a$: RUN
6990 STOP
7000 PLOT 24,175: DRAW 0,-175
7010 PRINT AT 0,0; PAPER 1;"Y"
7020 PRINT AT 21,31; PAPER 1;"X"
7040 RETURN
8000 BORDER 0: PAPER 0: INK 9: C
LS
8010 LET x=0
8020 RETURN
9000 FOR a=USR "a" TO USR "b"+7
9010 READ user: POKE a,user
9020 NEXT a: RETURN
9030 DATA 128,192,224,240,248,25
2,254,255
9040 DATA 1,3,7,15,31,63,127,255
```

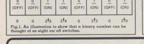


Fig.1. An illustration to show that a binary number can be thought of as eight on/off switches.

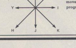
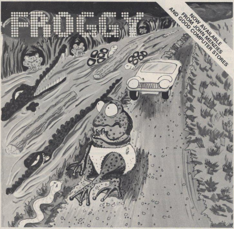


Fig.2. The controls of movement in the Sketch program.

The sophisticated Sketch program.

```
  30 LET x=0: LET y=0
  40 POKE 23658,8: LET f=0
  50 GO SUB 5000: PAUSE 0: GO SU
B 5000
  60 LET a$=INKEY$
  70 LET x=x+(a$="8" OR a$="I" O
R a$="K")-(a$="6" OR a$="Y" OR a
$="H")
  80 LET y=y+(a$="7" OR a$="6" O
R a$="8")-(a$="K" OR a$="H" OR a
$="J")
  90 IF IN 65022=247 THEN LET f=
1
 100 IF IN 32766=247 THEN LET f=
0
 110 IF IN 65022=251 THEN GO SUB
 1000
 120 IF IN 32766=251 THEN GO SUB
 1500
 130 IF IN 65022=253 THEN SAVE "
Screen"SCREEN$
 140 IF IN 57342=253 THEN GO SUB
 2000
 150 IF IN 57342=254 THEN GO SUB
 2500
 160 IF IN 65278=239 THEN CLS
 170 IF IN 65278=247 THEN GO SUB
 3000
 180 IF IN 63486=254 THEN GO SUB
 5000: PAUSE 0: PAUSE 0: GO SUB
5000
 190 IF IN 32766=239 THEN GO SUB
 3400
 230 PLOT OVER f;x,y
 240 PLOT OVER f;x,y
 250 GO TO 60
1000 REM DRAW
1010 INPUT "Input co-ordinates "
;x1;",";y1
1020 DRAW x1-x,y1-y
1030 LET x=x1: LET y=y1
1040 RETURN
1500 REM MOVE
1510 INPUT "Input co-ordinates "
;x1;",";y1
1520 LET x=x1: LET y=y1
1530 RETURN
2000 REM INK
2010 INPUT "Input new INK colour
";in
2020 INK in
2030 RETURN
2500 REM PAPER
2510 INPUT "Input new PAPER colo
ur ";pa
2520 PAPER pa
2530 DIM p?(1,704)
2540 PRINT AT 0,0; OVER 1; INK 8
;p$(1)
2550 RETURN
3000 REM CIRCLE
3010 INPUT "CIRCLE (x,y,r) x=";x
1;",y=";y1;",r=";r
3020 CIRCLE x1,y1,r
3030 RETURN
3500 REM BORDER
3510 INPUT "Input new BORDER col
our ";bo
3520 BORDER bo
3521 BEEP .5,0
3530 RETURN
5000 BEEP .5,0
5010 OVER 1
5020 PRINT AT 0,0;"MENU"
5030 PRINT "CONTROL KEYS (INCLUD
ING         DIAGONALS)"
5040 PRINT " 6  7  8
             ▛ ▀ ▜
            Y▌   ▐I
             ▙ ▄ ▟
            H  J  K"
5050 PRINT '"F - FLASHING CURSOR
"
5060 PRINT "N - DRAW MODE CURSOR
"
5070 PRINT "D - LINE DRAWN TO IN
PUT CO-ORDS."
5080 PRINT "M - MOVE CURSOR TO I
NPUT CO-ORDS"
5090 PRINT "S - SAVE SCREEN$"
5100 PRINT "V - CLS"
5110 PRINT "C - CIRCLE"
5120 PRINT "O - CHANGE INK"
5130 PRINT "P - CHANGE PAPER"
5140 PRINT "B - CHANGE BORDER"
5145 PRINT "1 - HELP (THESE INST
RUCTIONS"
5150 OVER 0
5160 RETURN
```



STOP PRESS: NEW SPECTRUM DOMINOES 16K/48K £4.95

ZUCKMAN
ZX81 (16K)

- *ALL MACHINE CODE (10K)
- *FOUR INDEPENDENT GHOSTS
- *HIGH-SCORE 'HALL OF FAME'
- *AUTHENTIC ARCADE ACTION
- *TITLE/DISPLAY MODE

ONLY £4.95 INC. P&P

★ ZX Spectrum ★
New New

FROGGY 16K or 48K

SPECTRUM VERSION OF ARCADE GAME WITH FULL ARCADE FEATURES:–

- *Fabulous Hi-Res Colour Graphics
- *Authentic Sound Effects + 3 Tunes
- *3-D Logs, Swimming & Diving Turtles
- *3 Lanes of Multi-coloured vehicles
- *On-screen Score, Hi-Score, Time-bar
- *Snake, Alligators and Baby Frog
- *'Top 5' High-Score initials table
- *Demonstration Game routine

Your home-sick frog must leap across a busy main road, then onto logs and turtles in the river to reach safety of the river bank frog-homes.

THIS MACHINE CODE GAME MUST BE SEEN TO BE BELIEVED!!

ONLY £5.95 INC. P&P

SCORE
HI-SCORE
FROGS
Time

FROGGY
ZX81 (16K)

- *MOVING CARS, LOGS, TURTLES
- *ALLIGATORS, DIVING TURTLES
- *FOUR 'SCREENS' OF ACTION
- *ALL ARCADE FEATURES
- *ENTIRELY MACHINE CODE

ONLY £4.95 INC. P&P

DJL SOFTWARE

DEPT GC3, 9 TWEED CLOSE, SWINDON,
WILTS SN2 3PU Tel: (0793) 724317

Trade enquiries welcome Export Orders: Please add £1.00 per tape airmail

Dealer enquiries: Tiger Distribution, 4 Victoria Road, Widnes, Cheshire. 051-420 8888.

ORDER FORM

Name ....

Address ....

....

....

Please send me the following ....

....

I enclose cheque/P.O. No .... for £ ....

SEND TO: DJL Software, Dept GC3, 9 Tweed Close, Swindon, Wilts SN2 3PU.

EARTH ATTACK

An exciting space adventure with just a sprinkling of machine code.

Written for the 16K and 48K Spectrum, this listing presents you with an exciting game which makes great use of machine code to save vital memory space. This program is divided into four main sections. These are:

1) Lines 1 to 180 contain all the initialisation and jumps to the user-defined graphics and machine code initialising routines. Since the amount of initialisation which needs to be done in this program is very large, there is quite a long pause after RUNning before the action starts – however, I shall explain later how this problem can be overcome.
2) Lines 200 to 999 contain the main game loop, including its servant subroutines (scoring, 'smart bomb' and high score routines). This loop calls all the machine code routines, and ties together their responses and any other relevant BASIC needed for the 'action' of the game.
3) Lines 1000 to 2400 contain the data and POKEing routines to set up first the user-defined graphics, then the machine code.
4) Lines 3000 to 3250 contain the three pages of instructions and game details which the computer cycles through between games.

The capital letters in quotes in the following lines should be entered in the Graphics mode, and when the program is RUN, produce the corresponding user-defined graphics to that key:

140, 180, 200, 230, 245, 701, 715, 725, 3140.

There are numerous other variables used in the program, but they are mostly control or dummy variables which are only used once and their use is fairly self-explanatory.

WELCOME TO THE MACHINE

There are three main machine code routines used in the program. The data for them starts at line 2000, and I have disassembled them for you in Figs. 3a, 4a and 4b.

First is the 'SMART BOMB' routine. This is 26 bytes long, and starts at address 32400. Using the attributes file of the Spectrum display, it changes the PAPER colour of every character position in the central third of the display (the part which the 'action' of this game utilises) to red. Whilst doing this it also calculates how many of the character positions previously contained yellow as their INK colour. This is the number which it returns to the variable, s, when it returns to BASIC, which is then multiplied by 10 to increase your score according to how many aliens your smart bomb destroyed.

Next comes the 'SIDE SCROLL' and 'BUILD CITY' routines, which actually run together, and are called by USR 32426. The height that the skyscrapers are built up to depends upon the variable, b, which is POKEd into location 32477. The routine operates once again only on the central third of the display. The reasons for this are:

▶

1) This is in keeping with the character of the game;
2) it speeds up the workings of code which has to operate on the thousands of bytes in the Spectrum's D_File, and
3) due to what is termed the curious construction of the display file, it makes manipulating one third of the screen at a time a fairly simple task, but crossing between the thirds rather complicated.

The last of these three main routines is the 'CHECK FIRING' routine. Actually it does rather more than its name suggests. Its action is as follows.

First the laser fire of the ship is drawn. At each step of this a check is made to see whether an alien has been hit or not.

If an alien has been hit, it checks whether this is a saucer or a monster, and adjusts the value of bc accordingly. It then creates a two stage explosion.

The routine then erases the laser fire, and returns to BASIC.

IN SHORT . . .

There are two other shorter machine code routines also included. The routine which starts at address 32350 and shown in Fig. 3b is a servant subroutine for the 'CHECK FIRING' code. When called from within another routine, it prints a character to the screen by POKEing the eight bytes which follow the call instruction to the character position, the start of which is specified by HL.

The routine starting at address 32300 produces the display of the city being destroyed at the end of the program, and is disassembled in Fig. 3c.

I mentioned before, the delay caused by the setting up of the machine code and user-defined graphics. Another problem with all this data is that it is an incredible waste of space, since once it has all been POKEd into the relevent addresses when the program was first run, it just sits there cluttering up your machine. And since the program as it stands uses very nearly all the space in a 16K machine, this means that, should you be tempted, you would find it hard to squeeze in many more lines before the Spectrum gives up the ghost.

The answer is, of course, to SAVE the code an user-defined graphics in a separate file on tape, and re-LOAD them each time you LOAD the program from tape. This is a very quick efficient method which really makes use of the space saving advantages which machine code can offer.

So, once your program has been typed in and you have RUN it once or twice to ensure that there are no errors in your typed-in version, you can erase every program line between 1000 and 2400 inclusive. Then type in the program lines shown in Fig. 1. Do not worry that

VARIABLES

The variables used, in order of their appearance, are:

s$	– Initials of the highest scorer.
h$	– The high score total.
s	– Current score.
bo	– INT s/1000 – used in calculating bonus bombs.
h	– Used throughout the program as a control variable in FOR. . .NEXT loops.
a$	– Displays the number of ships remaining (initially four).
l	– Number of laser shots.
sb	– The number of smart bombs ready for use.
x and y	– The co-ordinates of the ship.
p	– The screen byte directly in front of the ship (top byte of the character position).
rg	– The random variable used in selecting alien types.
b	– The variable used for POKEing into machine code (address 32477) to denote the height of the next building.
a	– The dummy variable for jumps to machine code.
p$	– Used in the high score routine to accept initials.

Fig. 3a. 'SMART BOMB' disassembled.

```
         ld, hl,ATTR FILE
         ld c,0
         ld b,255
LOOP     inc hl
         ld a,hl
         cp 6
         jrnz CHANGE
         inc c
         ld a,8
         ld (hl), a
         djnz LOOP
         ret
CHANGE   add a,16
         ld (hl),a
         djnz LOOP
         ret
```

Fig. 3b. 'CPOKE' disassembled.

```
         ld b,8
LOOP     pop de
         ld a,(de)
         inc de
         push de
         ld (hl),a
         ld de,256
         add hl,de
         djnz LOOP
         ret
```

Fig. 3c. End of game display disassembled.

```
LOOP5    ld hl,D FILE
LOOP4    ld c,32
         xor a
```

```
            ld a,7
LOOP2       ld hl,ATTR FILE
            ld b,255
LOOP1       ld (hl),a
            inc hl
            djnz LOOP1
            halt
            dec a
            jrnz LOOP2
            ld b,l
```

Fig. 4a. 'SIDE SCROLL' disassembled.

```
SIDE        ld b,64
SCROLL:     ld hl,D FILE
            ld de,32
            ld, a0
LOOP1       ld (hl),a
            add hl,de
            djnz LOOP1
            ld hl,ATTR FILE
            ld b,8
            ld de,32
            ld a,7
LOOP2       ld (hl),a
            add hl,de
            djnz LOOP2
            ld bc,2048
            ld de,D FILE
            ld hl,D FILE+1
            ldir
            ld bc 256
            ld de ATTR FILE
```

Fig. 4b. 'CHECK FIRING' disassembled.

```
            ld c,0
            ld hl,(DFCC)
            ld de,768
            add hl,de
            push hl
            ld b,5
LOOP1       ld a,(hl)
            cp 129
            jrz X1
            cp 127
            jrz X2
            ld (hl),255
            inc hl
            djnz LOOP1
            halt
            halt
            halt
            halt
            jr X3
X2          inc c
X1          push bc
            sbc hl,de
            push hl
            call CPOKE
```

```
LOOP3       ld a,(hl)
            srl a
            ld (hl),a
            dec hl
            dec c
            jrnz LOOP3
            ld a,71
            cp h
            jrnz LOOP4
            djnz LOOP5
            ret
```

```
            ld hl,ATTR FILE
            +1
            ldir
BUILD       ld b,0
CITY:       ld c,15
LOOP3       ld a,22
            rst 16
            ld a,c
            rst 16
            ld a,31
            rst 16
            ld a,16
            rst 16
            ld a,5
            rst 16
            ld a,17
            rst 16
            ld a,0
            rst 16
            ld a,148
            rst 16
            dec c
            djnz LOOP3
```

```
            DEFB 0,66,36,24,
            24,36,66,0
            halt
            halt
            pop hl
            push hl
            call CPOKE
            DEFB 153,90,
            16,199,227,8,
            90,153
            halt
            halt
            pop hl
            call CPOKE
            DEFB 0,0,0,0,
            0,0,0,0
            pop bc
            inc c
X3          pop hl
            ld a,5
LOOP2       ld (hl),0
            inc hl
            inc b
            cp b
            jrnz LOOP2
            ld b,0
            ret
```

some of these lines overwrite program lines that still exist – this is intended.

Now, find some room on a cassette somewhere, and SAVE the program and the two code files by typing:

```
RUN 2000
```

When re-LOADed, the program will automatically LOAD the two files and then RUN itself.

TIME FOR FUN!

And now, after all this effort, for the fun part – playing the game! Fig. 2 shows the keys used to control your rocket ship as it wings its way over a terrestrial cityscape. To add amusement there are a whole crop of little alien saucers out to get you. Although they don't actually shoot at you, there is a plentiful supply of them, and you can be destroyed in any of three ways: by crashing into them, by crashing into the city below you, or by accidentally moving into 'hyperspace' above. In fact, your little ship is not very sturdy at all, but you do have one weapon on your side – your lasers!

To start with you have 20 lasers, and your laser count is decremented by one each time you use them. You score 10 points for each saucer you hit, and 20 for each alien being (little yellow 'nasties' who stand on rooftops and are very hard to hit). Each time you bring the nose of your ship into contact with a green fuel base, your number of lasers is incremented by 20. It does not matter if you plough right through these fuel bases.

You get three smart bombs per 1,000 points scored. While the little bomb graphics are still showing beneath the city, each time you press the Space key the sky will turn red, and all visible aliens will be wiped out, and you will be given 10 points for each. Your ship will then accelerate out of the dangerous area.

The game ends when all five of your ships have been destroyed.

▶

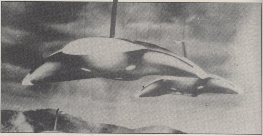


Fig.1.

```
  30 LOAD "mcode"CODE
  40 LOAD "chars"CODE USR "a"
2000 SAVE "attack" LINE 2050
2010 SAVE "mcode"CODE 32300,300
2020 SAVE "chars"CODE USR "a",6*
8
2030 PRINT "SAVEing finished": S
TOP
2050 RUN
```

Fig.2.

```
            EARTH ATTACK

 Use Keys:
            1-5......Up

            Q-T......Down

            6-0......Fire

            SHIFT....Extra Speed

            SPACE....Smart bombs

 PRESS "S" TO START
```

```
   1 CLEAR 32299
   2 BEEP .5,12: BEEP .5,24
  30 GO SUB 1000
  40 GO SUB 2000
  50 BORDER 0
  55 LET s$="..."
  60 LET hs=0
  65 LET s=0
  70 LET l=0
  80 LET bo=0
  85 GO TO 3000
  99 PAPER 0: INK 7: CLS : FOR n
=0 TO 20: BEEP .005,n: NEXT n
 100 LET x=9: LET y=5: LET b=1
 110 FOR n=0 TO 7: PRINT INK 1;A
T n,0;"█████████████████████████
███████": NEXT n
 120 LET s=0
 130 PRINT AT 20,0;"SCORE:";s
 140 LET a$="A A A A ": PRINT AT
 20,15;a$
 150 FOR n=16 TO 19: PRINT AT n,
0; INK 5;"██████████████████████
██████████": NEXT n
 160 LET l=20
 165 PRINT AT 0,0;"LASERS:";l;"
"
 170 PRINT AT 0,16;"HI-SCORE:";h
s
 180 PRINT AT 18,15; INK 2; PAPE
R 5;"DDD": LET sb=3
 200 LET x=x+(IN 64510<255)-(IN
63486<255): PRINT AT x,y;"A";
 210 LET p=PEEK (PEEK 23684+256*
PEEK 23685): IF p THEN GO SUB 50
0
 220 IF IN 61438<255 THEN IF l T
HEN GO SUB 400
 230 IF RND>.79 THEN PRINT AT RN
D*5+8,31; INK 6;"B"
 240 IF IN 32766<255 THEN GO SUB
 700
 245 LET rg=RND: IF rg>.96 THEN
PRINT AT 15-b,31; INK 4;("C" AND
 rg>.98)+(CHR$ 16+CHR$ 6+"F" AND
 rg<=.98): BEEP .005,10: BEEP .0
05,20
 255 LET b=INT (RND*3)+1
 260 PRINT AT x,y;" ": POKE 3247
7,b: LET a=USR 32426: IF IN 6527
8<255 THEN LET a=USR 32426
 270 GO TO 200
 400 REM FIRING SUBROUTINE
 405 LET l=l-1
 440 BEEP .005,30: LET a=USR 325
07: IF a THEN BEEP .01,7: BEEP .
01,0: GO TO 450
 442 PRINT AT 0,7;l;" "
 445 RETURN
 450 LET s=s+10*a
 460 PRINT AT 20,6;s;" ";AT 0,7;
l;" "
 470 RETURN
 500 IF p=60 THEN : LET l=l+20:
BEEP .01,20: GO TO 460
 501 BEEP .01,1: BEEP .01,2: BEE
P .01,4: BEEP .01,8: BEEP .01,16
: BEEP .01,8: BEEP .01,4: BEEP .
01,2: BEEP .01,1
 510 PRINT AT x,y; OVER 1;"#": B
EEP .01,0: BEEP .01,-.5: BEEP .0
1,-1
 515 IF a$="!" THEN GO TO 600
 520 LET a$=a$(3 TO ): IF a$=""
THEN LET a$="!"
 530 LET s=s-10: PRINT AT 20,15;
a$;"  "
 550 PRINT AT x,y; INK 1;"■": LE
T a=USR 32426: LET x=9
 560 BEEP .3,0: PRINT AT 20,6;s;
```

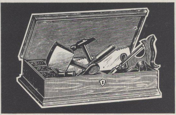
```
" ": RETURN
 600 PRINT AT 4,10; FLASH 1;"GAM
E OVER": FLASH 0: FOR n=1 TO 8:
LET a=USR 32300: BEEP .1,n: NEXT
 n: PAUSE 2: PAUSE 1: IF hs<s TH
EN LET hs=s: GO SUB 800
 610 GO TO 3000
 620 GO TO 100
 700 LET sb=sb-1
 701 IF INT (s/1000)>bo THEN LET
 sb=3: PRINT AT 18,15; INK 2; PA
PER 5;"DDD"
 702 IF sb<0 THEN RETURN
 710 LET s=s+10*USR 32400: BEEP
.5,7: BEEP .5,0: PRINT AT 20,6;s
;" ": PRINT AT 18,15+sb; INK 5;"
▓"
 715 IF INT (s/1000)>bo THEN LET
 sb=3: PRINT AT 18,15; INK 2; PA
PER 5;"DDD"
 716 LET bo=INT (s/1000)
 720 FOR n=0 TO 31: POKE 32477,I
NT (RND*3)+1: LET a=USR 32426
 725 PRINT AT x,y-1; PAPER 8;" A
": BEEP .01,n
 730 NEXT n: RETURN
 800 PAPER 0: INK 6: CLS
 810 PRINT "               EARTH ATTAC
K"
 820 LET a$="WELL DONE, YOU HAVE
 BEATEN THE  HIGH SCORE. PLEASE
ENTER YOUR    INITIALS NOW"
 830 PRINT ''
 840 FOR n=1 TO LEN a$: PRINT a$
(n);: BEEP .003,0: NEXT n
 850 PRINT ''TAB 5;
 860 LET s$=""
 862 PRINT "---";: FOR n=1 TO 3:
 PRINT CHR$ 8;: NEXT n
 865 FOR n=1 TO 3
 866 PRINT "?";
 870 PAUSE 0: LET p$=INKEY$: IF
LEN p$<>1 THEN GO TO 870
 880 PRINT CHR$ 8;p$;: LET s$=s$
+p$
 885 BEEP .1,n
 890 NEXT n
 900 PAUSE 50: RETURN
 999 STOP
1000 FOR n=1 TO 6: READ a$
1010 FOR p=0 TO 7: READ a
1020 POKE USR a$+p,a: NEXT p
1030 NEXT n
1040 DATA "a",BIN 11100000,BIN 1
10000,BIN 11111000,127,127,BIN 1
1111000,BIN 110000,BIN 11100000
1050 DATA "b",BIN 11000,BIN 1001
00,126,129,129,126,0,0
1055 DATA "c",BIN 111100,BIN 100
0010,129,BIN 1000010,60,60,60,60
1056 DATA "d",0,64,BIN 1011100,1
26,126,BIN 1011100,64,0
1057 DATA "e",255,BIN 10011001,B
IN 10011001,255,BIN 10011001,BIN
 10011001,255,255
1058 DATA "f",BIN 1011101,BIN 10
1010,BIN 110110,127,BIN 1010101,
BIN 1010101,BIN 10100,BIN 110110
1060 RETURN
2000 REM SMART BOMB
2010
2020 DATA 33,255,88,14,0,6,255,3
5,126,254
2030 DATA 6,32,7,12,62,8,119,16,
-12,201
2040 DATA 198,16,119,16,-18,201
2050
2060 REM SIDE SCROLL
2070
2080 DATA 6,64,33,0,72,17,32,0,6
2,0
2090 DATA 119,25,16,252,33,0,89,
6,8,17
2100 DATA 32,0,62,7,119,25,16,-4
,1,0
2110 DATA 8,17,0,72,33,1,72,237,
176,1
2120 DATA 0,1,17,0,89,33,1,89,23
7,176
2130
```

```
2140 REM [illegible]
2150
2160 DATA 6,0,14,15,62,22,215,12
1,215,62
2170 DATA 31,215,62,16,215,62,5,
215,62,17
2180 DATA 215,62,0,215,62,148,21
5,13,16,-26
2190 DATA 201
2200
2210 REM CHECK FIRING
2220
2230 DATA 14,0,42,132,92,17,0,3,
25,229,6,5,126,254,129,40,16,254
,127,40,11,54,255,35,16,-14,118,
118,118,118,24,47,12,197,237,82
2240 DATA 229,205,94,126
2250 DATA 0,66,36,24,24,36,66,0,
118,118,225
2252 DATA 229,205,94,126,153,90,
16,199,227,8,90,153,118,118,225
2260 DATA 205,94,126,0,0,0,0,0,0
,0
2265 DATA 0,193,12
2266 DATA 225,62,5,54,0,35,4,184
,32,-7,6,0,201
2270
2280 REM [illegible]
2300 FOR n=1 TO 26+50+31+92
2310 READ a: POKE 32399+n,a
2320 NEXT n
2330 DATA 62,7,33,0,89,6,255,119
,35,16,-4,118,61,32,-13,6,1,33,2
55,79,14,32,175,126,203,63,119,4
3,13,32,-8,62,71,188,32,-16,16,-
21,201
2340 FOR n=1 TO 39
2350 READ a: POKE 32299+n,a
2360 NEXT n
2370 DATA 6,8,209,26,19,213,119,
17,0,1,25,16,-11,201
2380 FOR n=1 TO 14
2390 READ a: POKE 32349+n,a: NEX
T n
2400 RETURN
3000 INK 7: PAPER 0: CLS
3010 LET a$=" DAVID HOLD "
3020 FOR n=1 TO LEN a$
3030 PRINT a$(n);
3040 NEXT n
3110 FOR n=0 TO 7: INK n: PRINT
AT 3,9; INK n;"EARTH ATTACK": BE
EP .1,n
3120 PRINT '"Use Keys:"'TAB 9;"1
-5......Up"''TAB 9;"Q-T......Dow
n"''TAB 9;"6-0......Fire"''TAB 9
;"SHIFT....Extra Speed"''TAB 9;"
SPACE....Smart bombs"
3121 PRINT '"PRESS 'S' TO START"
3122 NEXT n
3125 PAUSE 250
3126 IF INKEY$="s" THEN GO TO 99
3127 CLS
3130 PRINT "Earth Attack": PRINT
 : PRINT : PRINT
3140 LET a$="B........10 POINTS"
+CHR$ 13+CHR$ 13+"F........20 PO
INTS"+CHR$ 13+CHR$ 13+"C........
REFUELLING Stn"+CHR$ 13+CHR$ 13+
CHR$ 13+"PRESS 'S' TO START"
3150 FOR n=1 TO LEN a$
3160 PRINT a$(n);: BEEP .003,0:
NEXT n
3170 PAUSE 250: IF INKEY$="s" TH
EN GO TO 99
3180 CLS
3190 PRINT "EARTH ATTACK"
3200 PRINT "------------"
3205 PAUSE 50
3210 PRINT '''"HIGH SCORE=";HS:
BEEP .1,10: PAUSE 50
3220 PRINT '"LAST SCORE=";S: BEE
P .1,9: PAUSE 50
3230 PRINT ''"HIGH SCORE BY ";s$
: BEEP .1,8
3235 PRINT '''''"PRESS 'S' TO ST
ART"
3240 PAUSE 200: IF INKEY$="s" TH
EN GO TO 99
3250 GO TO 3000
```

Richard Sargent

TOOLBOX TRIO

Here are three BASIC utility programs.



The ability of the Spectrum to MERGE one program into another makes it very easy to tack one or more utility programs onto a BASIC program under development. Each utility program included in this article is a little longer than it need be because sensible prompts are included — however, these could be taken out if space is at a premium. Each program has been compacted to a reasonable number of lines so that it may be deleted fairly quickly when it is no longer required.

HEXADECIMAL/ DECIMAL CONVERSION

Once you have entered the two programs, you use the following inputs to access the utilities:

RUN 9000 enters the Hex to decimal conversion routine.

RUN 9011 enters the decimal to Hex conversion routine.

The programs will prompt for the number to be converted. Should you wish to change the direction of conversion you do not need to break and re-RUN the other program, you can simply input 'T' and '123456'. If you input '0', you stop the program.

PEEK AND POKE

RUN 9020 will PEEK a specified memory location. The program will ask you if you wish to read a single byte or a double byte, and then returns with a decimal number in the range 0 to 255 in the case of the former and 0 to 65535 in the case of the latter.

RUN 9030 will POKE a decimal number in the range 0 to 65535 into a specified memory location. Numbers greater than 255 will automatically be entered correctly into two consecutive memory locations.

ATTRIBUTE FINDER

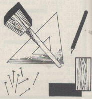
This program will be of use to anyone investigating a BASIC program when they only have a black and white television available. It may also be useful to anyone who has colour blindness. The BASIC program under investigation is stopped with the Break key and then RUN 9040 is entered.

Immediately the global attributes are displayed — the BORDER colour, the default PAPER colour, the default INK colour and BRIGHT/NORMAL mode — together with the attribute number that was set by the last PRINT statement before the break.

Then the user is prompted for a row and column co-ordinates until '33' is entered, which will stop the program.

The Attribute finder program itself obtains the global attributes by PEEKing three locations in the Spectrum workplace. This is done in line 9041. Local attributes are picked up using ATTR(R1,C1) in line 9048.

The attribute number, held in N, is broken up into its component parts by the subroutine starting at line 9050. Notice the use of computed GOSUBs in lines 9052 and 9053 which pick up the names of the colours.



```
9000 REM Hex to Dec Conversion
9001 PAPER 7: INK 0: CLS : PRINT
 "CAPS LOCK ON PLEASE."'"0 ABORT
s."'"T FOR DEC TO HEX."
9002 INPUT "HEX";H$: LET T=0: LE
T D=1: IF H$="0" THEN GO TO 9019
9003 IF H$="T" THEN GO TO 9011
9004 FOR P=LEN (H$)-1 TO 0 STEP
-1
9005 LET C=CODE (H$(D TO D)): LE
t D=D+1: IF C>=48 AND C<=57 THEN
 LET C=C-48: GO TO 9008
9006 IF C>=65 AND C<=70 THEN LET
 C=C-55: GO TO 9008
9007 PRINT "Try again": GO TO 90
02
9008 LET T=T+C*16↑P
9009 NEXT P
9010 PRINT "HEX ";H$;" = ";T;" D
EC": GO TO 9002
9011 PRINT "0 ABORTS": PRINT "12
3456 FOR HEX TO DEC."
9012 LET A$="0123456789ABCDEF"
9013 INPUT "DEC ";Q: IF Q=0 THEN
 GO TO 9019
9014 IF Q=123456 THEN GO TO 9002
9015 IF Q>65535 OR Q<0 THEN PRIN
T "Try again": GO TO 9013
9016 LET D=INT (Q/4096): LET R=Q
-4096*D: LET E=INT (R/256): LET
R=R-256*E: LET H=INT (R/16): LET
 L=R-16*H
9017 PRINT "DEC ";Q;" = ";
9018 PRINT A$(D+1 TO D+1);A$(E+1
 TO E+1);A$(H+1 TO H+1);A$(L+1 T
O L+1);" HEX": GO TO 9013
9019 STOP
```

Program 1. Hex-Dec and Dec-Hex conversions.

```
9020 PRINT "PEEKER"
9021 INPUT "SINGLE OR DOUBLE (S/
D)":A$: IF A$="0" THEN STOP
9022 INPUT "DEC ADDR = ";N: IF N
=0 THEN STOP
9023 PRINT "Address ";N;" contai
ns ";: IF A$="D" OR A$="d" THEN
PRINT PEEK N+256*PEEK (N+1): GO
TO 9021
9024 PRINT PEEK N: GO TO 9021
9030 PRINT "POKER"
9032 INPUT "DEC ADDR = ";N: IF N
=0 THEN STOP
9034 INPUT "DEC VALUE =";V: IF V
<=255 THEN POKE N,V: GO TO 9038
9035 POKE N,V-256*INT (V/256): P
OKE N+1,INT (V/256)
9038 PRINT "Address ";N;" contai
ns ";V: GO TO 9032
```

Program 2. PEEK and POKE with ease.

```
9040 REM ATTRIBUTE FINDER
9041 LET B=INT ((PEEK 23624)/8):
 LET A=PEEK 23693: LET T=PEEK 23
695: GO SUB 9063
9042 LET N=A: GO SUB 9050
9043 GO SUB 9055+B
9044 PRINT AT 19,0; INK 0;"GLOBA
L =";A;" LOCAL=";T;" BORDER=";B;
" ";AT 20,0;"PAPER=";P$;" INK=";
I$;" ";R$;AT 21,0;"Press 0 to qu
it, else continue.": PAUSE 0: IF
 INKEY$="0" THEN STOP
9045 GO SUB 9063
9046 PRINT AT 19,0; INK 0;"LOCAL
 ATTRIBUTE=";AT 21,0;"Enter 33 t
o quit."
9047 INPUT "ROW NUMBER IS ";R1:
INPUT "COLUMN NUMBER IS ";C1: IF
 R1<0 OR R1>21 OR C1<0 OR C1>31
THEN STOP
9048 LET N=ATTR (R1,C1): GO SUB
9050
9049 PRINT AT 19,16; INK 0;N;AT
20,0;"PAPER=";P$;" INK=";I$;" ";
R$: GO TO 9047
9050 LET R$=" NORMAL": IF N>=128
 THEN LET N=N-128
9051 IF N>=64 THEN LET N=N-64: L
ET R$=" BRIGHT"
9052 LET P=INT (N/8): LET I=N-P*
8: GO SUB 9055+P
9053 LET P$=C$: GO SUB 9055+I
9054 LET I$=C$: RETURN
9055 LET C$="BLACK ": RETURN
9056 LET C$="BLUE  ": RETURN
9057 LET C$="RED   ": RETURN
9058 LET C$="PURPLE": RETURN
9059 LET C$="GREEN ": RETURN
9060 LET C$="CYAN  ": RETURN
9061 LET C$="YELLOW": RETURN
9062 LET C$="WHITE ": RETURN
9063 FOR R=19 TO 21
9064 FOR C=0 TO 31: PRINT AT R,C
; PAPER 7; INK 0;" ": NEXT C: NE
XT R: RETURN
```

Program 3. Attribute finder.

David Spencer

SPECTRUM BREAKOUT

Fancy a spot of demolition? Well, just read on...

Based on the 16K ZX Spectrum, this game takes just over 6K of memory and makes full use of the sound and colour available on the machine.

The wall itself comprises three different coloured layers, each with 30 bricks. When 60 bricks have been knocked out, a new wall appears and the scores for each layer increases.

When you begin to play the game, you have five lives. However, each time you miss the ball with your bat, you will lose a life. The game ends when you have cleared all five walls or you have lost all five of your lives. All the appropriate sound effects have been included for when you hit the ball with your bat and when you knock out a brick from the wall.

When the game is not being played, the program cycles through a sequence of two explanation pages followed by a demonstration game in which the computer guides the bat. The highest score and the relevant player's name is displayed on screen at this time.

Here is a list of all the variables used in the program.

HS – Highest score.
N$. – Name of the highest scorer.
F – General FOR . . . NEXT counter.
FP – Play flag (1 for the demonstration game).
P – X co-ordinate of the bat.
W – Wall number.
S – Score.
ES – Temporary INPUT string.
X,Y – Co-ordinates of the ball.
D-
X,DY – Movement vector for the ball.
FL – Flag for sound effects.

FT – Flag for hitting top of screen.
FA – Flag for hitting bat.
L – Lives.
A – Variable to indicate which layer has been hit.
J – Set to 0 for game, 8 for demonstration game.
C – Code of character at ball position.
H – Number of bricks knocked out.
P1 – Required position of bat during demonstration game.
DP – Direction of bat movement during demonstration game.
YT – Temporary variable used in determining P1.

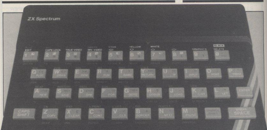
The listing for the program. Spectrum Breakout.

```
   1 REM **BREAK-OUT
   3 BORDER 0: LET hs=0: LET n$=
""
   5 GO SUB 2000: PRINT AT 21,2;
  10 PRINT AT 19,5; FLASH 1; BRI
GHT 1;"PRESS ANY KEY TO PLAY"
  12 FOR f=1 TO 1600: IF INKEY$<
>"" THEN GO TO 15
  13 NEXT f: GO TO 3000
  15 BEEP .5,0: LET fp=0: LET j=
0
  20 GO SUB 3000: GO TO 1000
  99 REM **KEYBOARD SCAN**
 100 IF fp=1 THEN GO TO 5000
 105 IF INKEY$<>"z" AND INKEY$<>
"m" THEN RETURN
 110 PRINT AT 21,p;"  "
 120 IF INKEY$="z" AND p>1 THEN
 LET p=p-1
 130 IF INKEY$="m" AND p<29 THEN
 LET p=p+1
 140 PRINT AT 21,p; INK 2;"▀▀":
RETURN
 199 REM **END OF WALL**
 200 PAUSE 100: CLS : IF fp=1 TH
EN GO TO 5
 203 LET w=w+1: PRINT AT 4,4; PA
PER 5;"Wall no. ";w-1;" destroye
d."
 205 FOR f=0 TO 10: BEEP .2,f: N
EXT f
 210 IF w=6 THEN GO TO 230
 220 PAUSE 50: PRINT AT 6,4; PAP
ER 4; INK 7;"Now go onto wall ";
w: PAUSE 150: GO SUB 9007: GO TO
 1000
 230 PRINT AT 6,3; PAPER 4; INK
0; FLASH 1;"WELL DONE, "; FLASH
0;"you have destroyed all the wa
lls"
 240 PRINT AT 9,4;"Your score wa
s ";s
 249 REM **HIGH. SCORE ROUTINE**
 250 IF s<hs THEN PRINT AT 11,4;
"Therefore you did not beat  the
 highest score of ";hs;" by ";n$
: GO TO 600
 260 IF s=hs THEN PRINT AT 11,4;
"And you equaled the highest sco
re": INPUT "ENTER YOUR NAME "; L
INE e$: IF e$="" THEN GO TO 260
 265 IF s=hs THEN LET n$=n$+" &
"+e$: GO TO 600
 270 IF s>hs THEN PRINT AT 11,4;
"And you beat the highest    sco
re!": INPUT "ENTER YOUR NAME ";
LINE n$: IF n$="" THEN GO TO 270
 280 LET hs=s: GO TO 600
 299 REM **ALL LIVES LOST**
 300 FOR f=0 TO 7: PAPER f: CLS
: PAUSE 20: NEXT f
 310 PRINT AT 4,11; INVERSE 1;"G
AME OVER": PAUSE 50: GO TO 240
 600 FOR f=15 TO -5 STEP -1: BEE
P .3,f: NEXT f
 610 PAUSE 200: GO TO 5
 999 REM **MAIN BALL ROUTINE**
1000 PRINT AT y,x;"o"
1002 IF fp=1 THEN GO SUB 5050
1003 IF fl=1 AND fp=0 THEN BEEP
.05,a*4: LET fl=0
1005 GO SUB 100: LET y=y+dy: LET
 x=x+dx: IF x=1 OR x=30 THEN LET
 dx=-dx: PRINT AT y-dy,x+dx;" ":
 GO TO 1009
1006 IF ft=1 THEN LET ft=0: GO T
O 1009
1007 PRINT AT y-dy,x-dx;" ": IF
fa=1 THEN GO SUB 140: LET fa=0
1008 IF fp=1 THEN GO SUB 5100
1010 IF y>6 AND y<21 THEN GO TO
1000
1020 IF y<7 THEN GO TO 1050
1029 REM **HIT BAT ROUTINE**
1030 IF x=p OR x=p+1 THEN LET dy
=-dy: LET x=x+1-(2*(x>=29)): LET
 y=y-1: LET fa=1: BEEP .1*(fp=0)
,0: GO TO 1005
1035 IF INKEY$<>"" THEN GO TO 10
35
1040 BEEP .75*(fp=0),30: LET l=l
-1: PAUSE 50: IF l=0 THEN GO TO
300
1045 PRINT AT 21,p;"  ": GO SUB
9050: GO TO 1000
1049 REM **HIT WALL ROUTINE**
1050 LET a=ATTR (y,x)-48-j: LET
 c=CODE SCREEN$ (y,x): IF a=1 AN
D y>1 AND c<>95 THEN GO TO 1000
1055 IF a=1 AND c<>95 THEN LET d
y=-dy: GO TO 1000
1070 IF c=95 THEN GO TO 1100
1075 LET fl=1
1080 LET s=s+(((a-2)*10)*w): LET
 h=h+1: PRINT AT 0,9;s: IF h=60
THEN GO TO 200
1090 LET dy=-dy: GO TO 1000
1100 LET x=x+(2*(RND>.5)-1): LET
 dy=-dy: LET ft=1: GO TO 1005
1999 REM **1st EXP. PAGE**
2000 PAPER 7: CLS : INK 0
2010 PRINT AT 2,10; PAPER 6;"BRE
AK-OUT";AT 2,9; INK 2; OVER 1;"_
__________"
2020 PRINT AT 4,0; PAPER 6;"
 The object of the game is  to h
it bricks out of the wall byboun
cing the ball off the bat.     "
2030 PRINT AT 9,0; PAPER 4;"
When you have knocked out 60 bri
cks a new wall appears. The game
 ends after 5 walls,or when you
runout of lives.               "
2040 PRINT AT 15,0; PAPER 2;"
 You move the bat left with  the
 Z key,and right with the M.  "
2050 RETURN
2999 REM **2nd EXP. PAGE**
3000 CLS : PRINT AT 2,10; PAPER
6;"BREAK-OUT";AT 2,9; INK 2; OVE
R 1;"___________"
3010 PRINT AT 4,1; PAPER 4;"
You start with 5 lives        "
3020 PRINT AT 6,1; PAPER 5;"
The highest score is ";hs;"by "
;n$;".";AT 7,0; PAPER 7;" "
3030 PRINT AT 9,1; PAPER 6;" On
the first wall the scores ";AT 1
0,1; PAPER 6;"are as follows:-"
3040 PRINT AT 11,7; PAPER 6;"10
for the bottom layer.";AT 12,7;"
20 for the middle layer.";AT 13,
7;"30 for the top layer.     "
3050 PRINT AT 15,1; PAPER 3;" On
 the next wall the scores  ";AT
16,1;"are twice the above and so
 on."
3060 PRINT AT 19,5; FLASH 1; BRI
GHT 1;"PRESS ANY KEY TO PLAY"
3070 PRINT AT 21,2; INK 3;"© Jul
y 1982, David M. Spencer."
3080 FOR f=1 TO 1600: IF INKEY$<
>"" THEN GO TO 15
3090 NEXT f
3999 REM **DEMO. GAME**
4000 LET j=8: LET fp=1: LET p1=0
: LET dp=0: GO SUB 9000: GO TO 1
000
4100 IF x<15 AND dx=1 THEN LET p
1=x+14: GO TO 4140
4105 IF x>15 AND dx=-1 THEN LET
p1=x-14: GO TO 4140
4110 IF dx=1 THEN LET yt=7+(32-x
): LET p1=30-(22-yt)
4120 IF dx=-1 THEN LET yt=7+x: L
ET p1=22-yt
4140 IF p1=0 THEN LET p1=1
4145 IF p1>=30 THEN LET p1=29
4150 IF p1=p THEN LET dp=0: RETU
RN
4160 IF p>p1 THEN LET dp=-1: RET
URN
4170 LET dp=1: RETURN
4999 REM **AUTO BAT MOVE**
5000 IF INKEY$<>"" THEN GO TO 15
5010 IF p=p1 OR p+1=p1 THEN RETU
RN
5020 PRINT AT 21,p;"  ": LET p=p
+dp: GO TO 140
5050 IF dy=-1 THEN LET p1=15: GO
 TO 4150
5060 RETURN
5100 IF y=7 AND dy=1 THEN GO TO
4100
5110 RETURN
8999 REM **INIT. ROUTINE**
9000 LET s=0: LET w=1: LET l=5
9007 LET h=0
9010 BORDER 0: PAPER 6: INK 1: C
LS
9015 IF fp=1 THEN PAPER 7: CLS
9020 FOR f=0 TO 21: PRINT PAPER
0;AT f,0;"█";AT f,31;"█": NEXT f
9030 FOR f=1 TO 3: PRINT AT 1,f*
7;"_____": NEXT f
9040 FOR f=1 TO 30: PRINT INK 5;
AT 4,f;CHR$ 143; INK 4;AT 5,f;CH
R$ 143; INK 3;AT 6,f;CHR$ 143: N
EXT f
9050 PRINT AT 0,3;"Score ";s;AT
0,22;"Lives ";l
9055 IF fp=1 THEN GO TO 9080
9060 PRINT AT 18,5; FLASH 1; PAP
ER 0; INK 5;"PRESS ANY KEY TO ST
ART"
9070 PAUSE 0: PRINT AT 18,5; PAP
ER 6;"                     "
9080 LET p=15: LET x=INT (RND*20
)+2: LET y=20: LET dx=-1+(2*(RND
>.5)): LET dy=-1: LET fa=0: LET
ft=0: LET fl=0: GO TO 140
```

Dave Hawke

ALTERNATIVE CHARACTERS

Getting tired of the Spectrum's character set? Why not type in this useful utility?



This routine for the 16K or 48K Spectrum transfers the character set from ROM to RAM and then modifies the characters. The system variable, CHARS, can then be made to point to the new character set.

When RUN, the program asks you to enter the Spectrum memory size and takes approximately two minutes to complete its task. The screen display should then show how to obtain the normal and alternate character sets and how to SAVE the bytes where the characters are located on tape.

POKING AROUND

For the 48K Spectrum, to obtain the new character set you have to POKE 23607,250. To SAVE the program, you should use the following:

SAVE name CODE 64000,1024

If you have a 16K machine, you should POKE 23607,123 to get to the new character set. And to SAVE the program, you should use:

SAVE name CODE 31486,1024

To return to the normal character set on either machine, you should POKE 23607,60.

```
  10 REM Alternate Character Set

  20 BORDER 6: PAPER 6: INK 0: C
LS
  30 LET A=PEEK 23606+256*PEEK 2
3607
  40 PRINT AT 21,2;"Enter Spectr
um Memory Size."
  50 INPUT "16 or 48 ";RAM
  60 IF RAM<>16 AND RAM<>48 THEN
 GO TO 50
  70 PRINT AT 20,6;"Please Wait
2 Minutes         Character Set be
ing Generated."
  80 IF RAM=16 THEN LET C=31488
  90 IF RAM=48 THEN LET C=64000
 100 REM >>Transfer and Modify<<
 110 FOR N=C TO C+1024: LET B=PE
EK A: POKE N,B
 120 IF B/4=INT (B/4) THEN POKE
N,B+2
 130 IF B/8=INT (B/8) THEN POKE
N,B+4
 140 IF B/16=INT (B/16) THEN POK
E N,B+8
 150 IF B/32=INT (B/32) THEN POK
E N,B+16
 160 IF B/64=INT (B/64) THEN POK
E N,B+32
 170 IF B=66 THEN POKE N,B+32
 180 IF B=0 THEN POKE N,0
 190 LET A=A+1: NEXT N: BEEP .1,
25
 200 REM >>Display Results<<
 210 CLS : POKE 23607,C/256
 220 PRINT AT 2,9;"POKE 23607,";
C/256
 230 PRINT '" To Obtain this Cha
racter Set."
 240 POKE 23607,60
 250 PRINT AT 8,9;"POKE 23607,60
"
 260 PRINT '" To Return to Norma
l Characters."
 270 PRINT AT 14,0;"SAVE:-SAVE n
ame CODE ",C;",1024"
 280 PRINT ''"This Basic Program
may be NEWed Leaving Sub-Routin
e above Ramtop"''"       Press any
Key to STOP"
 290 PAUSE 0: CLEAR C: STOP
 300 SAVE "alt char" LINE 1
```

STATISTICAL ANALYSIS

Check out your vital statistics with this great package.

ZX Spectrum
GRAPHICS
BLACK
DELETE
SPACE

This program, here presented for the 48K Spectrum (but with notes that follow for the program to be converted for the 16K model) provides a thorough grounding in many of the statistical tests you may wish to use when analysing quantities of data.

Amongst the different options of analysis you will have at your fingertips are standard deviation, the coefficient of variables, the standard error of mean, the F-test, the mode, ▶

regression analysis, the correlation coefficient, a sample T-test and significance testing. There is also the option to create a graph of your analysis and COPY the screen at various times throughout your work without having to break into the program.

Also included in the program is a bar chart routine. This is a complex routine using minimum and maximum values to give a range, sort data into groups and print the resultant chart on the screen.

POINTS TO NOTE

Any page can be COPYed by pressing the 'c' key. However, long data lists will have to be converted prior to LPRINT.

Please note that the sorting routine is in BASIC and thus may take up to 15 minutes to sort out over 100 data pairs. Remember, patience is a virtue!

This program does not have any mathematical features unique to the Spectrum, and so could easily be converted to run on most other home computers.

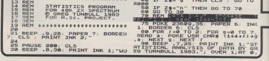
All you would have to note would be the POKE values, various BEEPs and colours used throughout.

The listing as it stands occupies around 22.4K of memory. However, if a note of all the instructions is made then these can be left out. Also, all the non-functioning lines can be removed along with the REM statements. You could also make use of multiple lines rather than the single statement lines I have used in some places.

In this manner, the program can be drastically reduced so that it will easily RUN on the 16K Spectrum without altering any of the program's operation.

PROGRAM STRUCTURE

Lines 1-35	Introduction to the program and the setting up of the screen.
Lines 40-190	Read the graphics, offer a choice of instructions, create the title, input data pairs number (DIMensions two sets of arrays), input x and y data, and print out the data.
Lines 200-595	Offer you the choice of altering any of the x and y data of any row, and then reprints the new data.
Lines 600-6030	Data for the graphics.
Lines 700-710	GOSUB 7000 to calculate all statistical values.
Lines 720-850	Provide a menu of choices — you can then choose which values are to be displayed or COPYd.
Lines 900-995	Descriptive statistics — this area provides you with the mean, standard deviation, coefficient of variables, standard error of means and the F-test.
Lines 1000-1090	PRINT/COPY data.
Lines 1100-1195	Sort data into descending order, and sorts it into B() and C() arrays. PRINTS ranked order.
Lines 1200-1290	Minimum, maximum and medians of data.
Lines 1300-1390	Statistical analysis — regression (b and c).
Lines 1400-1470	Correlation coefficient (r).
Lines 1500-1580	T-test and significance testing.
Lines 1600-1800	Draw a graph, axes names and values. Print data on graphs, checks and then draws a linear regression line through the data.
Lines 2000-3650	The bar chart routine, complete with instructions.
Lines 6000-6080	This routine asks you if you want to exit or run through the program again.
Lines 7000-7200	Calculate all the statistical values before the main routines — calculates means, CV, SD, SEM, F, t, DF, r, b and c.
Lines 8000-8730	Instructions on the use of all statistical tests and the means of the values.



```
  10 REM [illegible]
  11
  12 REM ************************
  13 REM
  14 REM    STATISTICS PROGRAM
  15 REM    FOR 48K ZX SPECTRUM
  16 REM    © GREG TUNBULL 1983
  17 REM    FOR M.Sc. PROJECT.
  18 REM
  19 REM ************************
  20
  21 BEEP .9,28: PAPER 7: BORDER
 7: CLS : PRINT INK 2;"
  24
  25 PAUSE 200: CLS
  30 BEEP .8,30: PRINT INK 1;"WO
ULD YOU LIKE INSTRUCTIONS?   (y
/n):": INPUT Z$
  40 IF Z$="y" THEN CLS : GO TO
8000
  50 IF Z$="n" THEN GO TO 70
  60 GO TO 30
  70 REM DATA INPUT, STORAGE,
PRINTING & READ GRAPHICS.
  75 POKE 23609,35: PAPER 6: INK
 1: BORDER 6: CLS
  80 FOR f=0 TO 2: FOR g=0 TO 7:
 READ a: POKE USR CHR$ (144+f)+g
,a: NEXT g: NEXT f
  90 BEEP .7,25: PRINT INK 1;"ST
ATISTICAL ANALYSIS OF DATA BY GR
EG TURNBULL 1983."; OVER 1;AT 0,
0;"_____________________________
__"; OVER 1;AT 1,0;"____________
__________      "''
 100 PRINT "TYPE IN YOUR TOTAL N
UMBER OF    DATA PAIRS (n):": IN
PUT r: PRINT "n=";r''
 110 LET A=r: LET R=A+1
 120 PRINT INK 1;"TYPE IN YOUR x
 AND y VALUES       (ENTER EACH ON
E):"''
 130 PRINT INK 3;"NOTE: IF YOU M
AKE A MISTAKE       WRITE IT DOWN
AND YOU WILL        BE ABLE TO ALT
ER IT LATER."
 140 DIM X(A): DIM Y(A): DIM B(R
): DIM C(R)
 150 PRINT : FOR n=1 TO A
 160 INPUT x: LET X(n)=x: LET B(
n)=x
 170 INPUT y: LET Y(n)=y: LET C(
n)=y
 180 PRINT INK 1;"n:";n;"  ";"x:
";X(n),"y:";Y(n)
 190 NEXT n: PRINT
 200 REM CHOICE OF ALTERING DATA.
 210 BEEP .8,30: PRINT INK 2;"WO
ULD YOU LIKE TO ALTER          AN
Y DATA? (y/n):": INPUT Q$: CLS
 220 IF Q$="y" THEN GO TO 250
 230 IF Q$="n" THEN GO TO 700
 240 GO TO 210
 250 PRINT "WOULD YOU LIKE TO CH
ANGE          x OR y VALUES? (x/y)
:": INPUT V$: CLS
 260 IF V$="y" THEN GO TO 410
 270 IF V$="x" THEN GO TO 290
 280 GO TO 250
 290 BEEP .8,30: PRINT INK 1;"CH
ANGE AN x VALUE:"; OVER 1;AT 0,0
;"___________________  "''
 300 PRINT "INPUT THE DATA ROW N
UMBER          FOR THE CHANGE (n):"
: INPUT N: CLS
 305 IF N>A THEN GO TO 300
 310 PRINT "INPUT THE x VALUE YO
U WANT         FOR THIS ROW (x):":
INPUT W: CLS
 320 LET X(N)=W: LET B(N)=W
 330 PRINT "DO YOU WANT TO ALTER
 ANY MORE   x VALUES? (y/n):": I
NPUT Q$: CLS
 340 IF Q$="y" THEN GO TO 290
 350 IF Q$="n" THEN GO TO 370
 360 GO TO 330
 370 PRINT "DO YOU WANT TO ALTER
 ANY           y VALUES? (y/n):": I
NPUT Q$: CLS
 380 IF Q$="y" THEN GO TO 410
 390 IF Q$="n" THEN GO TO 530
 400 GO TO 370
 410 BEEP .8,30: PRINT INK 1;"CH
ANGE A y VALUE:"; OVER 1;AT 0,0;
"___________________  "''
 420 PRINT "INPUT THE DATA ROW N
UMBER          FOR THE CHANGE (n):"
: INPUT N: CLS
 425 IF N>A THEN GO TO 420
 430 PRINT "INPUT THE y VALUE YO
U WANT         FOR THIS ROW (y):":
INPUT V: CLS
 440 LET Y(N)=V: LET C(N)=V
 450 PRINT "DO YOU WANT TO ALTER
 ANY MORE   y VALUES? (y/n):": I
NPUT Q$: CLS
 460 IF Q$="y" THEN GO TO 410
 470 IF Q$="n" THEN GO TO 490
 480 GO TO 450
 490 PRINT "DO YOU WANT TO ALTER
 ANY           x VALUES? (y/n):": I
NPUT Q$: CLS
 500 IF Q$="y" THEN GO TO 290
 510 IF Q$="n" THEN GO TO 530
 520 GO TO 490
 530 PRINT INK 3;"NEW DATA VALUE
S:"; OVER 1;AT 0,0;"____________
_____"''
 540 FOR n=1 TO A
 550 PRINT "n:";n;"  ";"x:";X(n)
,"y:";Y(n)
 560 NEXT n
 570 PRINT : PRINT INK 2;"WOULD
YOU LIKE TO ALTER          ANY FU
RTHER DATA? (y/n):": INPUT Q$: C
LS
 580 IF Q$="y" THEN GO TO 250
 590 IF Q$="n" THEN GO TO 700
 595 GO TO 570
 600 REM DATA FOR GRAPHICS.
 610 DATA 0,126,66,32,24,32,66,1
26
 620 DATA 255,0,195,102,60,60,10
2,195
 630 DATA 255,0,195,102,60,24,48
,224
 650 REM COPY DATA TO PRINTER.
 660 INPUT "PRESS ENTER TO CONTI
NUE:";W$
 670 IF W$="c" THEN COPY : CLS :
 RETURN
 680 IF W$="" THEN CLS : RETURN
 690 GO TO 650
 700 PRINT PAPER 1; INK 6; FLASH
 1;"PLEASE WAIT."'': PRINT PAPER
 5; INK 1; FLASH 0;"THE COMPUTER
 IS CALCULATING      THE STATISTI
CAL VALUES."''
 710 GO SUB 7000: CLS
 720 REM MENU OF CHOICES.
 730 CLS : BEEP .8,30: PRINT INK
 1;"WHICH TYPE OF ANALYSIS
   WOULD YOU LIKE?"''
 740 PRINT INK 2;"A) DESCRIPTIVE
 STATISTICS:"; OVER 1;AT 3,3;"___
_______________________"'': PRINT I
NK 1;"TYPE IN '1' FOR MEAN,SD,CV
(%),  SEM & F-TEST:": PRINT "TYP
E IN '2' TO PRINT THE DATA": PRI
NT "TYPE IN '3' FOR MIN,MAX & ME
DIAN"''
 750 PRINT INK 2;"B) STATISTICAL
 ANALYSIS:"; OVER 1;AT 10,3;"____
______________________"'': PRINT INK
1;"TYPE IN '4' FOR REGRESSION AN
D  CORRELATION": PRINT "TYPE IN
'5' FOR 2-SAMPLE t-TEST"''
 760 PRINT INK 2;"C) GRAPHS:"; O
VER 1;AT 16,3;"_______"'': PRINT
INK 1;"TYPE IN '6' FOR A LINE GR
APH": PRINT "TYPE IN '7' FOR A B
AR CHART"'': PRINT INK 3;"TYPE I
N '0' TO EXIT:"
 770 INPUT W$: IF W$="" THEN GO
TO 720
 780 LET W=VAL W$: CLS
 790 IF W=0 THEN GO TO 6000
 800 IF W>0 AND W<4 THEN GO TO 8
00+W*100
 810 IF W>4 AND W<7 THEN GO TO 1
000+W*100
 820 IF W=4 THEN GO TO 1300
 830 IF W=7 THEN GO TO 2000
 840 IF W>7 THEN GO TO 720
 900 REM START OF DESCRIPTIVE S
STATISTICS.
 905
 910 REM MEANS, SD, CV(%), SEM &
F-TEST.
 915 BEEP .8,30: PRINT INK 3;"DE
SCRIPTIVE STATISTICS:"; OVER 1;A
T 0,0;"_________________________"''
 920 LET s3=INT (s3*10↑5+.5)/(10
↑5): LET s4=INT (s4*10↑5+.5)/(10
↑5)
 930 LET v1=INT (v1*10↑3+.5)/(10
↑3): LET v2=INT (v2*10↑3+.5)/(10
↑3)
 940 LET s5=INT (s5*10↑5+.5)/(10
↑5): LET s6=INT (s6*10↑5+.5)/(10
↑5)
 950 LET h1=s3↑2: LET h2=s4↑2
 955 IF h1>h2 OR h1=h2 THEN LET
h3=h1/h2
 960 IF h2>h1 THEN LET h3=h2/h1
 965 LET h3=INT (h3*10↑4+.5)/(10
↑4): LET h4=A-1
 970 PRINT "N=";A,"","(B,C)=";"(
";p;",";q;")": PRINT "S.D of x="
;s3: PRINT "CV(%) of x=";v1: PRI
NT "S.E.M for x=";s5''
 975 PRINT "SD of y=";s4: PRINT
"CV(%) of y=";v2: PRINT "S.E.M f
or y=";s6''
 980 PRINT INK 3;"F-TEST:"; OVER
 1;AT 12,0;"_______"'': PRINT INK
 1;"A TEST TO COMPARE THE
   PRECISION OF x WITH THAT OF
y.": PRINT "THE F VALUE=";h3: PR
INT "FOR DF1 & DF2 OF:";h4: PRIN
T "THIS VALUE CAN BE LOOKED UP I
N  F-TABLES TO FIND A PROBABILIT
Y  (P) VALUE FOR SIGNIFICANCE
  TESTING."
 990 GO SUB 650: GO TO 720
1000 REM PRINT DATA.
1010 BEEP .8,30: PRINT INK 2;"DA
TA:"; OVER 1;AT 0,0;"____"''
1020 IF A<21 THEN GO TO 1060
1030 LPRINT "DATA."; OVER 1;AT 0
,0;"_____"'': FOR n=1 TO A
1040 LPRINT "n:";n;"  ";"x:";X(n
),"y:";Y(n)
1050 NEXT n: GO TO 1090
1060 FOR n=1 TO A
1070 PRINT "n:";n;"  ";"x:";X(n)
,"y:";Y(n)
1080 NEXT n
1090 PRINT : GO SUB 650: GO TO 7
20
1100 REM [illegible]
1110 PRINT PAPER 1; INK 7; FLASH
 1;"PLEASE WAIT."'': PRINT PAPER
 6; INK 1; FLASH 0;"THE COMPUTER
 IS RANKING THE DATA"''"DATA PAI
RS LEFT=";
1120 FOR J=A TO 1 STEP -1: FOR K
=1 TO A
1130 LET l=0: IF B(K+1)>B(K) THE
N LET l=B(K): LET B(K)=B(K+1): L
ET B(K+1)=l
1140 LET m=0: IF C(K+1)>C(K) THE
N LET m=C(K): LET C(K)=C(K+1): L
ET C(K+1)=m
1150 NEXT K: PRINT AT 4,16;"  "
;AT 4,16;J-1: PAUSE 9: PRINT AT
4,16;"  ";AT 4,16;J-1: NEXT J
1160 CLS : BEEP .8,30: PRINT INK
 3;"RANKED DATA:"; OVER 1;AT 0,0
;"____________"''
1165 FOR l=1 TO A
1170 PRINT "n:";l;"  ";"x:";B(l)
,"y:";C(l)
1180 NEXT l
1190 PRINT : GO SUB 650
1200 REM MIN, MAX AND MEDIANS.
1205 PRINT INK 3;"MIN, MAX, MEDI
ANS:"; OVER 1;AT 0,0;"__________
_______"''
1210 LET s=A/2: IF s=INT s THEN
GO TO 1230
1220 LET M1=B(s): LET M2=C(s): G
O TO 1240
1230 LET w=A/2: LET v=A/2+1: LET
 M1=(B(w)+B(v))/2: LET M2=(C(w)+
C(v))/2
1240 PRINT "MEDIAN OF x=";M1: PR
INT "MINIMUM VALUE OF x IS ";B(A
): PRINT "MAXIMUM VALUE OF x IS
";B(1)''
1250 PRINT "MEDIAN OF y=";M2: PR
INT "MINIMUM VALUE OF y IS ";C(A
): PRINT "MAXIMUM VALUE OF y IS
";C(1)''
1260 LET M1=INT (M1*10+.5)/10: I
F M1=INT (p*10+.5)/10 OR M1+.1=I
NT (p*10+.5)/10 OR M1-.1=INT (p*
10+.5)/10 THEN PRINT "median of
x approx' equal to   mean of x
indicating normally   distribute
d x data."''
1270 LET M2=INT (M2*10+.5)/10: I
F M2=INT (q*10+.5)/10 OR M2+.1=I
NT (q*10+.5)/10 OR M2-.1=INT (q*
10+.5)/10 THEN PRINT "median of
y approx' equal to   mean of y
indicating normally   distribute
d y data."''
1280 GO SUB 650: GO TO 720
1300 REM START OF STATISTICAL
ANALYSIS.
1305
1310 REM REGRESSION AND
CORRELATION.
1320 BEEP .8,30: PRINT INK 3;"RE
GRESSION AND CORRELATION:"; OVER
 1;AT 0,0;"_____________________
______"''
1330 LET E=INT (E*10↑5+.5)/(10↑5
): LET D=INT (D*10↑5+.5)/(10↑5):
 LET U=INT (U*10↑5+.5)/(10↑5)
1340 PRINT "Ax=";t1,"Ax*x=";t3,"
Ay=";t2,"Ay*y=";t4,"Ax*y=";t5
1350 PRINT "n=";A,"","(B,C)=";"(
";p;",";q;")"'': PRINT PAPER 1;
INK 6;"For y=b*x+c"'': PRINT "b=
";D,"","c=";E'': PRINT INK 3;"CO
R. COEFF', r=";U''
1360 IF D>.75 AND U>.75 THEN PRI
NT "b and r are >.75 indicating
a   strong positive linear
    correlation between x and y.
"''
1370 IF D<-.75 AND U<-.75 THEN P
RINT "b and r are <-.75 indicati
ng a  strong negative linear
     correlation between x and
y."''
1375 IF D<.75 AND U>.75 THEN PRI
NT "correlation coefficient (r)<
.75 indicating a strong positive
    relationship between x and y
."
1380 IF U=0 THEN PRINT "r=0 this
 indicates that x and y are tota
lly uncorrelated."
1390 GO SUB 650: GO TO 720
1500 REM t-TEST AND DF.
1510 BEEP .8,30: PRINT INK 3;"2-
SAMPLE t-TEST:"; OVER 1;AT 0,0;"
______________"''
1515 LET T=INT (T*10↑5+.5)/(10↑5
)
1520 PRINT PAPER 4; INK 1;"For D
F=",u2''
1530 PRINT INK 2;"THE VALUE OF t
=";T''
1540 PRINT INK 1;"LOOK UP YOUR t
 VALUE IN t-TABLESFOR THE ABOVE
DF, AND FIND THE  PROBABILITY VA
LUE (P)           FOR SIGNIFICAN
CE TESTING."''
1550 GO SUB 650: GO TO 720
1600 REM START OF GRAPHICAL
SECTION.
1601
1605 REM DRAW LINE GRAPH.
1610 BEEP .8,32: PRINT INK 3;"A
LINE GRAPH OF x,y:"; OVER 1;AT 0
,0;"_____________________"''
1620 PRINT INK 1;"HAVE YOU BEEN
VIA 3) MEDIANS    ROUTINE? (y/n)
:": INPUT I$: IF I$="n" THEN GO
TO 720
1630 IF NOT I$="y" THEN CLS : GO
 TO 1600
1640 PRINT "TYPE IN THE NAME OF
THE x-AXIS": INPUT A$
1650 PRINT "TYPE IN THE NAME OF
THE y-AXIS": INPUT B$: CLS
1560 LET F=B(1): LET G=C(1)
1670 IF F>26 THEN LET F=26
1680 IF G>18 THEN LET G=18
1690 FOR O=17 TO (8*F+40): PLOT
O,9: NEXT O
1700 FOR M=9 TO (8*G+22): PLOT 1
7,M: NEXT M
1710 PRINT AT 21,3;A$: PRINT AT
```

```
18-G,2;B$: PRINT AT 19-G,0;G: PR
INT AT 21,2.5+F;F: PRINT AT 21,1
;"0"
1720 FOR n=1 TO A
1730 IF X(n)>=26 OR Y(n)>=18 THE
N PRINT AT 19,3;"DATA IS TOO LAR
GE": GO SUB 650: GO TO 720
1735 BEEP .02,30: INK 2: OVER 1:
 CIRCLE 8*X(n)+17,8*Y(n)+9,2
1740 NEXT n
1750 IF p>10 OR q>10 THEN PRINT
AT 19,3;"MEAN VALUES ARE TOO LAR
GE": GO SUB 650: GO TO 720
1760 FOR z=1 TO 20: BEEP .05,50-
z: NEXT z: INK 1: PLOT 17,8*E+9:
 DRAW p+16*P,q+16*P: PAUSE 200
1770 GO SUB 650: GO TO 720
2000 REM BAR CHART.
2010 BEEP .8,30: PRINT INK 3;"BA
R CHART."; OVER 1;AT 0,0;"______
__"''
2020 PRINT INK 1;"HAVE YOU BEEN
VIA THE MEDIANS   ROUTINE? (y/n)
 THIS IS BECAUSE  THE MAXIMUM AN
D MINIMUM VALUES  ARE NEEDED FOR
 THE BARCHART."'': INPUT z$
2040 IF NOT z$="y" THEN GO TO 72
0
2060 PRINT INK 1;"WOULD YOU LIKE
 A BAR CHART OF   THE x OR y DAT
A?                TYPE IN x OR y
                  OR PRESS ENTER
 TO RETURN:"'': INPUT z$: CLS
2070 IF z$="x" THEN GO TO 2095
2080 IF z$="y" THEN GO TO 2500
2085 IF z$="" THEN GO TO 720
2090 GO TO 2060
2095 PRINT FLASH 1;"PLEASE WAIT.
"'' FLASH 0;"THIS MAY TAKE SOME
TIME!"
2096 REM CALCULATE x VALUES.
2100 LET minx=B(A): LET maxx=B(1
): DIM R(15): DIM L(16): LET div
x=(maxx-minx)/15
2110 FOR B=1 TO 15: FOR n=1 TO A
2120 LET P9=0: LET R9=minx+divx
2125 IF B=15 THEN LET R9=R9+.000
001
2130 IF X(n)>=minx AND X(n)<R9 T
HEN LET P9=P9+1: LET R(B)=R(B)+P
9: LET L(B)=L(B)+P9
2140 NEXT n
2150 LET minx=minx+divx
2160 NEXT B
2165 PRINT "NEARLY FINISHED."
2170 FOR J=1 TO 15: FOR K=1 TO 1
5
2180 LET l=0: IF L(K+1)>L(K) THE
N LET l=L(K): LET L(K)=L(K+1): L
ET L(K+1)=l
2190 NEXT K: NEXT J
2200 LET highx=L(1): LET scalex=
1
2210 IF highx>18 THEN LET scalex
=scalex+1: LET highx=highx-18: G
O TO 2210
2250 GO TO 3000
2500 PRINT FLASH 1;"PLEASE WAIT.
"'' FLASH 0;"THIS MAY TAKE SOME
TIME!"
2505 REM CALCULATE y VALUES.
2510 LET miny=C(A): LET maxy=C(1
): DIM S(15): DIM M(16): LET div
y=(maxy-miny)/15
2520 FOR B=1 TO 15: FOR n=1 TO A
2530 LET P9=0: LET R9=miny+divy
2535 IF B=15 THEN LET R9=R9+.000
001
2540 IF Y(n)>=miny AND Y(n)<R9 T
HEN LET P9=P9+1: LET S(B)=S(B)+P
9: LET M(B)=M(B)+P9
2550 NEXT n
2560 LET miny=miny+divy
2570 NEXT B
2575 PRINT "NEARLY FINISHED."
2580 FOR J=1 TO 15: FOR K=1 TO 1
5
2590 LET l=0: IF M(K+1)>M(K) THE
N LET l=M(K): LET M(K)=M(K+1): L
ET M(K+1)=l
2600 NEXT K: NEXT J
2610 LET highy=M(1): LET scaley=
1
2620 IF highy>18 THEN LET scaley
=scaley+1: LET highy=highy-18: G
O TO 2620
2630 GO TO 3500
3000 REM DRAW BAR CHART OF x.
3010 IF scalex=1 THEN GO TO 3030
3020 FOR B=1 TO 15: LET R(B)=INT
 (R(B)/scalex+.5): NEXT B
3030 BEEP .25,20: CLS : DIM J$(1
8): PRINT "INPUT THE NAME OF THE
 HORIZONTAL(x) AXIS:": INPUT I$
3040 PRINT "INPUT THE NAME OF TH
E VERTICAL  (y) AXIS:": INPUT J$
: CLS
3050 BEEP .8,30: PLOT 16,16: DRA
W 0,143: PLOT 16,16: DRAW 239,0:
 PRINT AT 0,4;"A BAR CHART OF x.
"; OVER 1;AT 0,4;"______________
__": PRINT AT 19,1;"0": PRINT AT
 20,2;B(A): PRINT AT 20,28;B(1):
 PRINT AT 21,6;I$: FOR z=0 TO 17
: PRINT AT z+4,0;J$(z+1): NEXT z
: PRINT AT 2,0;18*scalex
3055 LET sp=2
3060 FOR B=1 TO 15
3065 IF R(B)=0 THEN LET sp=sp+2:
 GO TO 3110
3070 FOR J=1 TO R(B)
3080 PRINT INK 2;AT 20-J,sp;"■":
 PRINT INK 2;AT 20-J,sp+1;"■"
3090 NEXT J
3100 LET sp=sp+2
3110 NEXT B
3150 GO SUB 650: CLS : GO TO 206
0
3500 REM DRAW BAR CHART OF y.
3510 IF scaley=1 THEN GO TO 3530
3520 FOR B=1 TO 15: LET S(B)=INT
 (S(B)/scaley+.5): NEXT B
3530 BEEP .25,20: CLS : DIM K$(1
8): PRINT "INPUT THE NAME OF THE
 HORIZONTAL(x) AXIS:": INPUT I$
3540 PRINT "INPUT THE NAME OF TH
E VERTICAL  (y) AXIS:": INPUT K$
: CLS
3550 BEEP .8,30: PLOT 16,16: DRA
W 0,143: PLOT 16,16: DRAW 239,0:
 PRINT AT 0,4;"A BAR CHART OF y.
"; OVER 1;AT 0,4;"______________
__": PRINT AT 19,1;"0": PRINT AT
 20,2;C(A): PRINT AT 20,28;C(1):
 PRINT AT 21,6;I$: FOR z=0 TO 17
: PRINT AT z+4,0;K$(z+1): NEXT z
: PRINT AT 2,0;18*scaley
3560 LET sp=2
3570 FOR B=1 TO 15
3580 IF S(B)=0 THEN LET sp=sp+2:
 GO TO 3630
3590 FOR J=1 TO S(B)
3600 PRINT INK 2;AT 20-J,sp;"■":
 PRINT INK 2;AT 20-J,sp+1;"■"
3610 NEXT J
3620 LET sp=sp+2
3630 NEXT B
3650 GO SUB 650: CLS : GO TO 206
0
6000 REM EXIT.
6010 PRINT "ARE YOU SURE? (y/n):
": INPUT Y$: CLS
6020 IF Y$="n" THEN GO TO 720
6030 IF Y$="y" THEN GO TO 6050
6040 GO TO 6010
6050 PRINT "WOULD YOU LIKE TO RU
N          THE PROGRAM AGAIN? (
y/n):": INPUT Y$: CLS
6060 IF Y$="y" THEN RESTORE 600:
 GO TO 70
6070 IF Y$="n" THEN BEEP .9,20:
PRINT INK 2;"O.K. BYE!": STOP
6080 GO TO 6050
7000 REM CALCULATION OF SUM OF
```

```
TOTALS, MEANS, SD, CV, SEM, REG,
COR, DF & t
7010 LET t1=0: LET t2=0: LET t3=
0: LET t4=0: LET t5=0: LET S1=0:
 LET S2=0
7020 FOR h=1 TO A
7030 LET t1=t1+X(h): LET t2=t2+Y
(h): LET t3=t3+X(h)↑2: LET t4=t4
+Y(h)↑2: LET t5=t5+X(h)*Y(h)
7040 NEXT h
7050 LET q=t2/A: LET q=INT (q*10
↑3+.5)/(10↑3): REM MEAN(y)
7060 LET p=t1/A: LET p=INT (p*10
↑3+.5)/(10↑3): REM MEAN(x)
7070 LET D=(t5-(t1*t2/(h-1)))/(t
3-(t1↑2/(h-1))): REM REG(b)
7080 LET U=(t5-(t1*t2/(h-1)))/SQ
R ((t3-(t1↑2/(h-1)))*(t4-(t2↑2/(
h-1)))): REM COR(r)
7090 LET E=q-D*p: REM REG(c)
7100 FOR i=1 TO A
7110 LET s1=s1+(X(i)-p)*(X(i)-p)
: LET s2=s2+(Y(i)-q)*(Y(i)-q)
7120 NEXT i
7130 LET s3=SQR (s1/(i-2)): LET
s4=SQR (s2/(i-2)): REM SD(x&y)
7140 LET v1=s3/p*100: LET v2=s4/
q*100: REM CV(x&y)
7150 LET s5=s3/SQR (i-1): LET s6
=s4/SQR (i-1): REM SEM(x&y)
7160 LET u2=2*A-2: REM DF
7170 IF q>p THEN LET T=(q-p)/SQR
 ((2/(u2*A))*(t3+t4-(t1↑2/A)-(t2
↑2/A))): GO TO 7190
7180 LET T=(p-q)/SQR ((2/(u2*A))
*(t3+t4-(t1↑2/A)-(t2↑2/A))): REM
 t-TEST
7190 RETURN
8000 REM INSTRUCTIONS.
8010 BEEP .8,30: PRINT INK 2;"IN
STRUCTIONS ON STATISTICS:"; OVER
 1;AT 0,0;"______________________
________"''
8020 PRINT INK 1;"THIS PROGRAM A
LLOWS YOU TO INPUTTWO SETS OF DA
TA OF ANY LENGTH  FOR A 48K SPEC
TRUM. THE PROGRAM CAN BE MODIFIE
D TO RUN ON A 16K SPECTRUM IF AL
L NON-ESSENTIAL    LINES, REM STA
TEMENTS AND        INSTRUCTIONS A
RE LEFT OUT OF THELISTING. NOTE:
 BOTH SETS OF DATAMUST BE OF
EQUAL LENGTH, THE TOTAL NUMBER
OF DATA PAIRS      BEING ENTERED
AS n"''
8030 GO SUB 650
8060 PRINT INK 3;"THE RAW DATA:"
; OVER 1;AT 0,0;"____________":
PRINT INK 1;"THIS WILL BE STORED
 IN TWO        SEPARATE ARRAYS, ON
E FOR          PRINTING AND ANALYS
IS, THE OTHERFOR RANKING (USED I
N MEDIANS).    YOU WILL THEN BE AB
LE TO ALTER    ANY OF THE DATA THA
T MAY HAVE     BEEN ENTERED INCORR
ECTLY. THE     STATISTICAL TESTS A
RE THEN        PERFORMED AND A MEN
U OF CHOICES PRINTED."''
8070 GO SUB 650
8100 PRINT INK 3;"DESCRIPTIVE ST
ATISTICS:"; OVER 1;AT 0,0;"______
_________________"'': PRINT INK
1;"THIS GIVES YOU THE MEANS
   (AVERAGES) OF THE TWO SETS OF
   DATA. IT ALSO GIVES YOU SOME
   MEASURES OF THE DISTRIBUTION
OF VALUES ABOUT THE MEANS.": PRI
NT INK 3;"THESE ARE A) STANDARD
DEVIATION:": PRINT OVER 1;AT 7,1
3;"__________________": PRINT IN
K 1;"THIS VALUE IS BOTH POSITIVE
 (+) AND NEGATIVE (-). AT THE 95
%    CONFIDENCE LIMIT LEVEL 95.4
% OF ALL VALUES SHOULD LIE WITHI
N +2 OR -2 SD FROM THE MEAN. SO
ANY  VALUE OUTSIDE THESE LIMITS
MUST BE CONSIDERED STATISTICALLY
       DIFFERENT FROM THE MEAN OF
       NORMALLY DISTRIBUTED DATA."
8120 PRINT : GO SUB 650
8150 PRINT INK 3;"B) COEFFICIENT
 OF VARIATION (CV)"; OVER 1;AT 0
,0;"_____________________________
______"'': PRINT INK 1;"THIS IS A
PERCENTAGE (%) VALUE  OFTERN USE
D TO EXPRESS THE      PRECISION
OF THE DATA.           IT RELATES
 BOTH THE SD AND MEAN BY: CV=SD/
MEAN*100%.  A LOW CV INDICATES
GOOD  PRECISION, AND AHIGH VALUE
 INDICATES POOR        PRECISION.
"''
8160 GO SUB 650
8190 PRINT INK 3;"C) STANDARD ER
ROR OF THE MEAN    (SEM):"; OVER
1;AT 0,0;"_______________________
_______"'': PRINT INK 1;"THIS IS
 AN ESTIMATE OF HOW CLOSETHE MEA
N OF YOUR SAMPLE IS TO   THE MEA
N OF THE POPULATION FROM WHICH I
T WAS TAKEN. HENCE THIS  VALUE M
AY NOT BE RELEVANT FOR   ALL SET
S OF DATA.               THE CAL
CULATION IS SEM=SD/SQR(n)SO AS S
AMPLE SIZE INCREASES,    THE SEM
 SHOULD DECREASE AS IT   APPROAC
HES THE 'TRUE' MEAN OF   THE POP
ULATION. (WHERE SEM=0)"''
8200 GO SUB 650
8230 PRINT INK 3;"F-TEST:"; OVER
 1;AT 0,0;"_______"'': PRINT INK
1;"THIS IS A TEST SOMETIMES USED
 TOCOMPARE THE PRECISIONS OF TWO
    SETS OF DATA. THE F VALUE IS
    CALCULATED FROM THE VARIANCE
    RATIO (VARIANCE=SD SQUARED) B
Y: F=(LARGER VARIANCE)/(SMALLER
    VARIANCE). THIS VALUE IS THE
    LOOKED UP IN F-TABLES FOR TWO
 DF(DEGREES OF FREEDOM) VALUES
    (IN THIS CASE BOTH VALUES ARE
    EQUAL AND ONLY ONE IS PRINTED
). FROM THE TABLES A PROBABILITY
    VALUE (P) IS FOUND AND THIS C
AN BE USED IN ASSESSING THE
    SIGNIFICANCE (IF ANY) BETWEEN
    THE TWO PRECISIONS. SEE ALSO
THENOTES ON THE t-TEST."
8250 PRINT : GO SUB 650
8280 PRINT INK 3;"PRINT DATA:";
OVER 1;AT 0,0;"___________": PRIN
T INK 1;"THE SECOND CHOICE ON TH
E MENU    ALLOWS YOU TO LIST THE
DATA.     THIS CAN BE COPIED BY P
RESSING   'c' INSTEAD OF ENTER, S
O ANY     TIME A HARD COPY OF THE
          INSTRUCTIONS OR THE ANA
LYSIS IS REQUIRED JUST PRESS 'c'
."''
8300 GO SUB 650
8320 PRINT INK 3;"MEDIANS & DATA
 SORTING:"; OVER 1;AT 0,0;"______
___________________": PRINT INK 1;
"THE THIRD CHOICE (AND FINAL PAR
TOF DESCRIPTIVE STATISTICS) WILL
 USE A SORTING ROUTINE TO RANK
 THE SPARE ARRAY DATA INTO
 DESCENDING ORDER (THIS MAY TAKE
 SOME TIME AS IT IS IN BASIC AND
 NOT MACHINE CODE!). ONCE THE
 DATA HAS BEEN RANKED IT WILL BE
 LISTED AND THE MAXIMUM, MINIMUM
 AND MEDIAN VALUES FOR x & y WIL
LBE DISPLAYED. THE MEDIAN IS A
 VALUE WHICH DIVIDES THE NUMBER
 OF OBSERVATIONS INTO TWO EQUAL
 PARTS. IN NORMALLY DISTRIBUTED
 DATA IT WOULD BE EQUAL, OR VERY
 CLOSE TO THE MEAN VALUE. THUS
 DIFFERENCES BETWEEN THE MEAN AN
DMEDIAN SHOWS LEFT OR RIGHT SKEW
 IN THE DATA."
8340 PRINT : GO SUB 650
8370 PRINT INK 3;"THE MODE:"; OV
```

```
ER 1;AT 0,0;"__________"'': PRINT
INK 1;"THIS IS THE MOST COMMONLY
       OCCURING VALUE IN ANY SET
 OF    DATA. IN THIS PROGRAM IT
CAN    EASILY BE SEEN IN THE LIS
T OF   RANKED DATA.IN NORMALLY
       DISTRIBUTED DATA THE MODE
, MEAN AND MEDIAN WOULD ALL BE
       (APPROXIMATELY) EQUAL. AN
Y LEFT OR RIGHT SKEW IN THE DATA
 WOULD MAKE THEM SIGNIFICANTLY
       DIFFERENT FROM EACH OTHER
."''
8380 GO SUB 650
8410 PRINT INK 3;"STATISTICAL AN
ALYSIS:"; OVER 1;AT 0,0;"_______
_______________"'': PRINT INK 1;"T
HESE CHOICES GIVE VARIOUS       C
ALCULATIONS ON THE DATA WHICH  W
OULD BE VERY TIME CONSUMING TO D
O MANUALLY. THEY ARE REGRESION,C
ORRELATION AND t-TEST."''
8420 GO SUB 650
8450 PRINT INK 3;"REGRESSION ANA
LYSIS:"; OVER 1;AT 0,0;"________
____________"'': PRINT INK 1;"THI
S IS AN ESTIMATE OF THE      ASS
OCIATION OF THE x DATA WITH  THE
 y DATA. LINEAR REGRESSION ISUSE
D, WHERE IF THE ASSOCIATION  IS
A  STRAIGHT LINE THEN IT WILLHAV
E THE FORMULAE y=b*x+c WHERE c=I
NTERCEPT ON THE y-AXIS AND   b=R
EGRESSION COEFFICIENT        (GR
ADIENT). IF THE ASSOCIATION  IS
TOTALLY LINEAR THEN b WILL BE+1
OR -1 DEPENDING ON THE SLOPE OF
THE LINE. STATISTICALLY      REG
RESSION IS USED TO DRAW A    BES
T-FIT LINE THROUGH SPREAD-OUTDAT
A."''
8460 GO SUB 650
8490 PRINT INK 1;"HERE b IS CLCU
LATED AND USED TO FIND THE VALUE
 OF c (BY c=C-b*B)THE LINE IS TH
EN DRAWN FROM c    THROUGH (B,C)
TO THE UPPER       LIMITS OF THE
DATA. THIS LINE    DRAWING METHOD
 IS THE ONE USED   ON THE PLOT A
GRAPH CHOICE.      NOTE: THE LINE
 ONLY WORKS FOR    POSITIVELY COR
RELATED DATA,      IF THE DATA IS
 NEGATIVELY        CORRELATED THE
 LINE WILL RUN OFFTHE SCREEN. TH
IS CAN BE AVOIDED IF PROGRAM LIN
ES 1750 & 1760 AREDELETED. SEE A
LSO GRAPH          INSTRUCTIONS."
''
8500 GO SUB 650
8530 PRINT INK 3;"CORRELATION CO
EFFICIENT:"; OVER 1;AT 0,0;"_____
_______________________"'': PRINT IN
K 1;"THIS IS ANOTHER EXPRESSION
OF    THE ASSOCIATION BETWEEN THE
 TWO SETS OF DATA. COMPLETE
      CORRELATION WILL GIVE AN r
VALUEOF +1 OR -1 DEPENDING ON TH
E     SLOPE OF THE LINE. IF THE D
ATA   IS TOTALLY UNCORRELATED THE
N r   WILL BE 0. CORRELATION AND
      REGRESSION ANALYSIS ARE USU
ALLY  USED IN CONJUNCTION AND ARE
      OFTERN DISPLAYED ON THE GRA
PHS   OF DATA."''
8540 GO SUB 650
8570 PRINT INK 3;"2-SAMPLE t-TES
T:"; OVER 1;AT 0,0;"____________
___"'': PRINT INK 1;"THERE ARE M
ANY TYPES OF t-TEST  ANALYSIS. T
HE ONE USED HERE IS  FOR TWO SAM
PLES OF EQUAL LENGTH.THE TEST IS
 USED TO DETERMINE   WHETHER THE
RE IS ANY SIGNIFICANTDIFFERENCE
BETWEEN THE TWO SETS OF DATA. TH
E t VALUE IS         CALCULATED
ALONG WITH THE       DEGREES OF
FREEDOM (DF) FOR THE DATA. FROM
THESE, STUDENT'S      t-TABLES CA
N BE USED TO FIND THEPROBABILITY
 VALUE (P) FOR        SIGNIFICANC
E TESTING."''
8580 GO SUB 650
8610 PRINT INK 3;"SIGNIFICANCE T
ESTING:"; OVER 1;AT 0,0;"________
_______________"'': PRINT INK 1;"T
HIS USES THE P VALUE (WHICH MAYB
E EXPRESSED AS A %) TO SEE IF  T
HERE IS ANY STATISTICAL        D
IFFERENCE BETWEEN THE TWO SETS O
F DATA FOR EXAMPLE P<=0.001    (
OR 0.1%) MEANS THAT THE DATA   A
RE STATISTICALLY THE SAME.     N
OTE: IF THE t VALUE EXCEEDS THET
ABULATED VALUES THEN THE       D
IFFERENCE IS SIGNIFICANT       (
ie: NOT LIKELY TO BE DUE TO    C
HANCE ALONE.)"''
8620 GO SUB 650
8650 PRINT INK 3;"THE LINE GRAPH
:"; OVER 1;AT 0,0;"_____________
_"'': PRINT INK 1;"THIS IS LIMIT
ED BY THE SCREEN     SIZE FOR THE
SPECTRUM. IT WILL    ONLY ALLOW (x
,y)<(26,20) TO BE    PRINTED, OR M
EANS<11. IT ALSO     ASSUMES POSIT
IVE LINEAR           CORRELATION B
ETWEEN x & y WHEN    IT DRAWS A LI
NE THROUGH THE       POINTS. HOWEV
ER AS MENTIONED      THIS COULD BE
 AVOIDED IF          NECESSARY BY
THE DELETION OF THERELEVANT LINE
S. IF A GRAPH OF     HIGH DATA IS
NEEDED, YOU WOULD    HAVE TO RE-EN
TER THE VALUES       AFTER DIVISIO
N BY A SUITABLE      FACTOR TO DEC
REASE THEIR SIZE."''
8660 GO SUB 650
8690 PRINT INK 1;"THE GRAPH ROUT
INE WILL ALLOW YOUTO NAME THE TW
O AXES, AND          SPECIFY THEIR
MAXIMUM LENGTHS.  THUS  THE ROUT
INE IS VERY USEFULFOR  VISUALISI
NG THE ACTUAL        RELATIONSHIPS
BETWEEN TWO SETS     OF DATA.  IT C
AN BE COPIED TO      THE PRINTER BY
 PRESSING 'c'."''
8700 GO SUB 650
8710 PRINT INK 3;"BAR CHART."; O
VER 1;AT 0,0;"__________"'': PRIN
T INK 1;"THIS ROUTINE ALLOWS YOU
 TO       CHOOSE A BAR CHART OF x
 OR y     VALUES. AGAIN IT IS LIM
ITED BY   THE SCREEN SIZE OF THE
SPECTRUM THE VALUES ARE SORTED I
NTO 15    DOUBLE-WIDTH COLUMNS BY
 A        DIVISION FACTOR CALCULA
TED FROM THE MAXIMUM AND MINIMUM
 DATA    VALUES. THUS YOU MUST G
O THROUGHCHOICE NUMBER 3 (MEDIAN
S) BEFOREUSING THE BAR CHART. TH
E DATA ISSORTED INTO AN ARRAY AC
CORDING  TO ITS PARTICULAR VALUE
. IF THE  COLUMN BECOMES TOO LARG
E (>17)   THEN THE COMPUTER WILL
SCALE     DOWN ALL THE VALUES AS
REQUIRED.THE USE OF THIS IS THAT
 IT WILL SHOW YOU IF THE DATA IS
 NORMALLYDISTRIBUTED THIS IS REQ
UIRED BY MANY STATS TESTS."
8720 GO SUB 650
8800 PRINT "PRESS r TO READ THES
E           INSTRUCTIONS AGAIN,
c TO COPY,   AND 'ENTER' TO RETUR
N TO         THE MAIN PROGRAM:":
INPUT W$
8810 IF W$="r" THEN CLS : GO TO
8000
8820 IF W$="" THEN CLS : GO TO 7
0
8830 IF W$="c" THEN COPY : CLS :
 GO TO 8700
8840 GO TO 8700
9000 REM THIS LISTING OCCUPIES
24.3 K.
```

MORSE TRAINER

Use your Spectrum to learn the Morse Code.

The program itself is a fairly straightforward listing being very user friendly and menu driven. In the test mode, the instructions for operation are displayed and a Morse character is BEEPed. It is then up to the user to INPUT a letter. Should you not provide the correct letter, you will be informed. You are given three opportunities to give the correct answer, after which the correct letter is displayed along with its relevant dots and dashes, and the letter is BEEPed in Morse code again.

A LOAD OF CODE

Mr Gilbert has used a number of useful POKEs in this program. Here is a brief explanation of these POKEs.

POKE 23609,X — This gives the keyboard click varying duration. When X=0, you hear the standard click. However, the value of X can be between 0 and 255, but when numbers greater than 10 are used, even though the feedback bleep is more noticeable, there is also a noticeable difference in the speed of the auto-repeat facility. This can be frustrating when editing long program lines. It also does not work with INKEY$.

POKE 23658,X — When this location is POKEd with a zero, it disengages the Caps Shift lock. When POKEd with an eight, it engages the Caps Shift lock. This can be very convenient as it saves checking for capital or lower case letters when a user is INPUTting information to a program.

POKE 23692, 1 — This will ensure that the program will not stop with the message, 'Scroll?,'. In this case, the PRINTing is started at co-ordinates 21,0 which would normally present you with that smashing little five letter word. Try omitting the POKE 23692 lines and have a look what happens!

In the FOR b NEXT b loop, i$ is attacked at the first letter and is turned into its Morse equivalent by READing the DATA the requisite number of times as set in line 95.

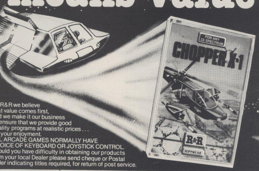
Once you get used to the program, the PAUSE times can be easily changed if you begin to find them a bit slow. Ex-ZX81 users who are dubious of utilising the PAUSE instruction due to the twitching screen syndrome need have no fears.

```
  1 REM Morse trainer
  2 REM © R.Gilbert MCMLXXXII
  5 POKE 23609,20
 10 DIM m$(7)
 15 DIM i$(250)
 20 INPUT "PITCH",P
 25 INPUT "SPEED",S
 35 LET H=S/100: LET T=S/300
 40 POKE 23658,8
 45 INPUT "X=alter speed/pitch
T=TEST","N=new phrase   ENTER=re
peat"; LINE A$
 50 IF A$="" THEN GO TO 70
 55 IF A$="X" THEN GO TO 20
 60 IF A$="T" THEN GO TO 300
 65 INPUT "ENTER Letter or Word
", LINE i$
 70 POKE 23692,2
 75 PRINT AT 21,0;"
                               "
 80 FOR b=1 TO LEN i$
 85 IF i$(b)="0" THEN LET m$="
      "
 90 IF i$(b)=" " THEN GO TO 40
 95 FOR r=1 TO (CODE i$(b)-64)
100 READ m$: NEXT r
105 RESTORE
110 PRINT m$( TO )
115 POKE 23692,2
120 GO SUB 200
125 NEXT b
130 GO TO 40
150 DATA "A.-","B-...","C-.-.",
"D-..","E.","F..-.","G--.","H...
.","I..","J.---","K-.-","L.-..",
"M--","N-.","O---","P.--.","Q--.
-","R.-.","S...","T-","U..-","V.
..-","W.--","X-..-","Y-.--","Z--
..""
200 FOR N=2 TO 6
210 IF m$(n)="-" THEN BEEP H,P
220 IF m$(n)="." THEN BEEP T,P
230 NEXT N
240 RETURN
300 CLS : PRINT "I will give yo
u a morse letter  and I want you
 to tell me what  it is in 3 att
empts                 Press ENTER
to escape ": PAUSE 100
305 LET c=0
310 FOR r=1 TO ((RND*26)+1)
315 READ m$
320 NEXT r
325 RESTORE
330 GO SUB 200
335 INPUT "WHAT WAS IT? "; LINE
N$
340 IF N$="" THEN CLS : GO TO 4
0
345 IF CODE N$>90 OR CODE N$<65
THEN GO TO 335
350 IF N$=m$(1) THEN PRINT : PR
INT : PRINT "Correct it was ";m$
355 IF N$=m$(1) THEN PAUSE 50:
GO SUB 200: PAUSE 200: GO TO 300
360 PRINT : PRINT : PRINT "Wron
g it was not ";N$;: PAUSE 75
365 LET c=c+1
370 IF c=3 THEN PRINT " It was
";m$: PAUSE 50: GO SUB 200: PAU
SE 120: GO TO 300
375 PRINT " TRY AGAIN": PAUSE 5
0
380 GO TO 330
```

R&R

SOFTWARE

means value



At R&R we believe
that value comes first,
and we make it our business
to ensure that we provide good
quality programs at realistic prices . . .
for your enjoyment.
ALL ARCADE GAMES NORMALLY HAVE
CHOICE OF KEYBOARD OR JOYSTICK CONTROL.
Should you have difficulty in obtaining our products
from your local Dealer please send cheque or Postal
Order indicating titles required, for return of post service.

CHOPPER X-1
(For any ZX Spectrum)

You command the Spectrum Chopper X-1 Gunship— Your mission is to repel invading forces and save the earth. An action packed original game in full machine code.

R.R.P. £5.50

GALAXY WARLORDS
(For any ZX Spectrum)

A fast action machine code space game in which you defend your position as Galaxy Warlord against attacking enemy starships. Both your patrol ship and rockets are steerable. The enemy never seem to give up! Full colour action graphics and sound for just

R.R.P. £5.50

JUNIOR ARITHMETIC
(For any ZX Spectrum)

An excellent educational program for 6–9 year olds. The fun way to learn and practice Addition, Subtraction, Multiplication and Division. Mistakes are corrected and explained, along with a running total of correct answers. Terrific value at just

R.R.P. £4.95

SPECTIPEDE
(For any ZX Spectrum)

Enjoy the thrill of this full machine code arcade game. Battle to keep the Spectipede at bay but watch out for a very unpredictable spider which is out to get you. For 1 or 2 players with top score and on-screen high score feature.

R.R.P. £5.50

GOLF
(For any ZX Spectrum)

Enjoy a game of golf in the comfort of your own home! With the choice of a 9 or 18 hole course and the challenge of Fairway, Rough, Trees, Bunkers, Water & Green. Displays are in realistic colour graphics with full score cards. No two games are alike!

R.R.P. £3.75

GNASHER
(For any ZX Spectrum)

The "famous" arcade maze game with fast machine code action, excellent graphics, colour and sound. Eat up all the dots before the Ghosts eat you, or beat them to an energy pill which will allow you to eat them! Hours of addictive fun with on-screen scoring and high-score tape-save capability for just

R.R.P. £4.95

STAR TREK
(For 48K Spectrum only)

One of the original computer space games, but this program has many up-dated features normally only found on larger computer versions. Test yourself as Starship Commander as your mission takes you to the outer reaches of the Galaxy to protect Federation Space, the Starship and yourself! Excellent value at

R.R.P. £4.95

OTHER COMPUTERS?

Send S.A.E. for a full list of our growing range of software now available for ZX81 and ORIC-1Computers.

DEALER ENQUIRIES WELCOME
R&R Software Ltd.
5 Russell Street, Gloucester GL1 1NE.
Tel (0452) 502819

R&R
SOFTWARE

CALLING ALL PROGRAMMERS . . .

WANTED New, Quality Software.
Send us your latest Program
for evaluation – NOW.

AVAILABLE FROM LEADING DEALERS NATIONWIDE

Andrew Cole

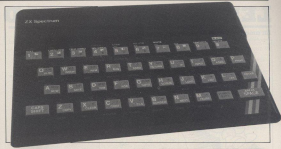
ALIENS

Can you defend the Earth from the invading aliens?

In this game for your 16K Spectrum, you command a missile base which you can manoeuver along the surface of the Earth, using the 'p' key to move right and the 'o' key to move left. To fire a missile you have to press the 'q' key.

The aliens, in their usual unfriendly way, re-energise at different levels above the Earth one at a time. It is your job to shoot them out of the sky before they get enough ships out of hyperspace to destroy the Earth. Once they have 10 ships positioned over the surface, they will be in a position to do this. So, your mission, should you choose to accept it, is to keep the number of aliens in the sky down to less than 10.

You can only fire one missile at a time on-screen, but if you find it too difficult to play at one level, you can always choose another.



```
  10 GO SUB 1000
  20 GO SUB 1200
  30 REM screen setup
  35 BORDER 0
  40 CLS : PRINT AT 20,2; PAPER
4;"
  "

  45 PRINT AT 21,2; PAPER 4;"
                       "
  50 FOR s=1 TO 20
  60 PLOT RND*254,RND*120+40
  70 NEXT s
  80 LET f1=0: LET l=15
  90 LET z=0: LET hits=0
 100 REM aliens
 110 POKE 23672,0
 120 LET t=PEEK 23672
 130 IF t>=80-(a*6) THEN PRINT A
T RND*17,RND*28+2; INK 2;"A": LE
T z=z+1
 140 IF z=10 THEN GO TO 400
 150 PRINT AT 21,21;z
 160 IF t>=80-(a*6) THEN POKE 23
672,0
 200 REM laser base
 205 LET l=l+(INKEY$="p")-(INKEY
$="o")
 210 LET l=l+(l=0)-(l=30)
 220 PRINT AT 19,l;" L ": PAUSE
4
 230 IF INKEY$="q" THEN GO TO 25
0
 240 IF INKEY$<>"q" THEN GO TO 1
20
 250 FOR c=18 TO 0 STEP -1
 260 PRINT AT c,l+1;"D": PAUSE 2
 270 PRINT AT c,l+1;" "
 275 IF ATTR (c-1,l+1)=58 THEN G
O TO 300
 280 NEXT c
 290 GO TO 130
 300 REM missile hit
 310 BEEP .1,z*2
 320 LET z=z-1: PRINT AT 21,21;z
 330 PRINT AT c-1,l+1; FLASH 1;"
E": PAUSE 50
 340 PRINT AT c-1,l+1;" "
 350 LET hits=hits+1
 360 PRINT AT 21,3;"HITS=";hits
 370 GO TO 130
 400 PRINT AT 4,8; INK 0; FLASH
1;"YOU'RE DEAD"
 410 PRINT AT 12,3;"Do you want
another game y/n"
 420 IF INKEY$="y" THEN GO TO 20
 430 IF INKEY$="n" THEN STOP
 440 GO TO 420
1000 FOR n=0 TO 7
1010 READ a: POKE USR "a"+n,a
1020 NEXT n
1030 DATA 129,126,219,126,60,60,
90,129
1040 FOR n=0 TO 7
1050 READ l: POKE USR "l"+n,l
1060 NEXT n
1070 DATA 24,24,24,24,60,126,255
,90
1080 FOR n=0 TO 7
1090 READ d: POKE USR "d"+n,d
1100 NEXT n
1110 DATA 16,16,16,16,16,16,56,4
0
1120 FOR n=0 TO 7
1130 READ e: POKE USR "e"+n,e
1140 NEXT n
1150 DATA 137,74,52,204,51,44,82
,145
1160 RETURN
1200 REM instructions
1210 PRINT AT 3,3;"You have to d
efend yourself     against the at
tacking aliens     who will re-en
ergise out of      hyperspace abo
ve the ground.      It is your jo
b to shoot the     aliens down us
ing your laser     base "
1220 PRINT AT 16,3;"PRESS ANY KE
Y TO CONTINUE"
1230 IF INKEY$=" " THEN GO TO 12
30
1240 PAUSE 0
1250 CLS : PRINT AT 3,3;"Your la
ser base is controlled   by keys
'o' for left,'p' for     right a
nd 'q' for fire.         If you
allow 10 aliens to be    present
 at the same time you    will be
 eliminated"
1260 PRINT AT 13,2;"SELECT SKILL
 LEVEL FROM 1-5"
1270 PRINT AT 15,5;"1=EASIEST  5
=HARDEST"
1280 LET a$=INKEY$
1290 INPUT "SKILL LEVEL= ";a$
1300 IF a$<"1" OR a$>"5" THEN GO
 TO 1280
1310 LET a=VAL a$
1320 RETURN
```

Clyde Bish

LEPRECHAUN'S GOLD

An amazing game for your 16K Spectrum.

This 3D maze game for the Spectrum does not have dinosaurs charging up on you but it does have a Green Goblin. He follows you about, sending an indecisive player back to the start, and giving you a pot of gold if you get to the centre.

INTERESTING POINTS

1. No maze plan is shown — There are two reasons for this: a) I think it makes the game too easy and b) the computer does not have a clue about the shape of the maze until play begins! — it all depends on the values of RND in line 210 and the moves a player makes. The maze does, however, once set, remain constant throughout the game (strings x$ and y$ ensure this).
2. The program does not use machine code — I feel that the speed of the Spectrum's PLOT and LIN DRAW commands are fast enough. And, by avoiding machine code, the structure of the program is more easily seen by those who like to dissect programs.
3. The problem of 16K being about 9K! — This has had to result in a compromise: a) The elements of the maze (1000-1710) use, for the most part, numbers and are placed at the beginning of the listing so that they are displayed quickly. b) The parts which can operate slower are put later and use variables and 'VALs' to save bytes. c) The UDGs are set separately and are SAVEd onto tape as bytes to be LOADed in by the main program.

NOTES ON ENTERING THE PROGRAM

1. As explained above the UDGs have to be set first. Type in the program starting at line 1 to line 20 and RUN this. This will set the UDGs above RAMTOP.
2. Now enter the rest of the program and SAVE it, followed by the UDG bytes using command:

Table 1. A breakdown of the program. Leprechaun's gold, line by line.

Lines	Description
190-280	: Logic controlling the patch through the maze (which is set up in line 8000 — see later). The basis is this:- a) 210 — If the next-but-one element is a dead-end ('7') then generate a random number. If this is less than 0.2 then the routine skips the dead-end, otherwise the dead-end is reached and the player is sent back to the beginning. b) 270 — When the player reaches this decision point again, if he or she chooses to take an alternate pathway (the original choice being held in string y$) then the dead end is avoided. c) 260 — If during a further attempt the player takes a turning different to the first attempt (excepting (b) above) then a dead end is reached and he or she is sent back to the beginning.
1000-1710	: The elements of the maze are displayed on the screen in 3D as if looking down the passage. There are three passage elements (1000 : left turn, 1100 : right turn and 1200 : straight on) and four junction elements (1300 : forward or right, 1400 : forward or left, 1500: left or right and 1600 : forward, left or

right) plus the dead-end at 1700. These are accessed in passage-junction pairs by line 200. Each element (except those at lines 1200 and 1700) are two part giving first a distant view followed by one with the player (see 4001) in a position to turn.

2000-2020 : If the player tries to walk into the passage wall this routine prevents him or her from doing so.

3000-3060 : This section invites a choice to be made (allowing only '5', '7 or '8'). If one is not made in about three seconds then the Green Goblin appears and sends the player back to the start.

4001 : This displays the player as a stick man with his hands on his hips.

4600 : A complex logic routine which, on the command '5' or '8', causes the figure to walk to the left or right and disappear around the corner of the passage.

4800 This causes the figure to move forwards on the command '7'.

5000 : This turns the figure to the left or right before walking (or bumping into the wall!).

6000-6015 : On LOADing, this is where the program begins. Instructions are given, variables are set, and the difficulty level (the number of elements in the maze) is chosen.

7000-7999 : The winning sequence. The gold is reached (much to the chagrin of the Green Goblin) and the player is invited to play again. It should be noted that the program can be stopped at any junction by pressing '0' in place of '5', '7' or '8' — see 3050.

8000 : Here the maze is set up as a string of numbers (representing the elements) held in x$. They are alternately 0-2 and 3-7, there not being two 7s following (or a junction would have two dead-end exits!).

9000 : This LOADs the user defined graphics (UDGs).

```
SAVE "maze" LINE 9000:
SAVE "maze" CODE
USR"a", 168
```

You will get the usual 'start tape then press any key' message. Do so, but don't walk away to make a cup of coffee! After about 45 seconds, the message will appear again. Don't stop the tape, just press any key and the UDG bytes will be saved.

To verify use the command:

```
VERIFY ""
```

After (hopefully) the 'OK' message appears stop the tape promptly, then verify the UDGs with :

```
VERIFY "" CODE.
```

3. To LOAD simply enter LOAD "maze". The main program will LOAD, then RUN itself from line 9000 which will LOAD in the UDGs (following on tape) before displaying the instructions, and away you go.

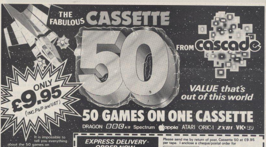
Hope you find the GOLD!

As a final note, the single or pairs of capital letters shown in the LISTing within quotes are the UDGs and should be entered in the G mode.

```
   1 FOR i=1 TO 19: READ p$: FOR
 n=0 TO 7: READ a: POKE USR p$+n
,a: NEXT n: NEXT i
   2 DATA "b",66,66,66,66,66,66,
66,195
   3 DATA "c",60,90,153,153,90,6
0,24,36
   4 DATA "d",24,24,24,24,24,24,
24,24
   5 DATA "e",16,16,16,16,16,16,
16,24
   6 DATA "f",8,8,8,8,8,8,8,24
   7 DATA "g",0,3,4,4,4,4,3,0
   8 DATA "h",0,192,32,32,32,32,
192,0
   9 DATA "i",3,5,9,17,33,65,1,2
  10 DATA "j",192,160,144,136,13
2,130,128,64
  11 DATA "k",2,4,4,8,8,16,16,24
  12 DATA "l",64,32,32,32,16,16,
16,24
  13 DATA "m",2,4,4,4,8,8,8,24
  14 DATA "n",64,32,32,16,16,8,8
,24
  15 DATA "o",124,124,124,124,12
4,56,68,68
  16 DATA "p",68,68,68,68,198,0,
0,0
  17 DATA "q",0,0,0,0,0,0,24,36
  18 DATA "r",36,24,60,60,60,24,
36,36
  19 DATA "s",36,36,0,0,0,0,0,0
  20 DATA "t",16,40,16,56,56,56,
40,40
 190 LET y$="
"
 200 FOR i=s TO y: GO SUB VAL x$
(i)*VAL "100"+VAL "1000": IF n>b
*v THEN GO TO VAL "200"
 205 LET i=i+s: GO SUB VAL x$(i)
*VAL "100"+VAL "1000": IF x$(i)=
"7" OR n>b*v THEN GO TO VAL "200
"
 210 IF y$(i)=" " AND x$(i+b)="7
" THEN LET y$(i)=b$: LET r=RND:
LET x$(i+b)=("8" AND r<VAL ".3")
+("7" AND r>=VAL ".3"): LET i=i+
(b AND r<VAL ".3"): GO TO VAL "2
80"
 220 IF y$(i)=" " AND x$(i+b)<>"
7" THEN LET y$(i)=b$: GO TO VAL
"280"
 230 IF y$(i)<>" " AND x$(i+b)="
8" AND y$(i)=b$ THEN LET i=i+b:
GO TO VAL "280"
 260 IF y$(i)<>" " AND y$(i)<>b$
 AND x$(i+b)<>"7" THEN GO SUB IN
T (RND*INT PI)*VAL "100"+VAL "10
00": GO SUB VAL "1700": GO TO VA
```

```
L "200"
 270 IF y$(i)<>" " AND y$(i)<>b$
 AND x$(i+b)="7" THEN LET y$(i)=
b$: LET i=i+b: LET x$(i)="8"
 280 NEXT i: GO TO 7000
1001 CLS : PLOT VAL "47",m+m: DR
AW VAL "128",a: PLOT VAL "47",VA
L "103": DRAW VAL "128",a: PLOT
x,a: DRAW m+m,m: DRAW a,VAL "119
": DRAW -u,u: PLOT VAL "240",a:
DRAW -o,m+m: DRAW a,VAL "71": DR
AW q,q
1010 LET p=w-s: GO SUB f: GO SUB
 e: IF n>b*v THEN RETURN
1015 IF b$<>"7" THEN LET p=w: GO
 SUB d: GO TO VAL "1000"
1020 LET c=s: LET p=m+b: GO SUB
h
1051 CLS : PLOT m,m+m: DRAW VAL
"159",a: PLOT m,VAL "103": DRAW
VAL "159",a: PLOT x,a: DRAW a,VA
L "167": DRAW -l,l: PLOT VAL "24
0",a: DRAW -o,m+m: DRAW a,VAL "7
1": DRAW q,q
1060 LET p=m+b: GO SUB f: GO SUB
 e: IF n>b*v THEN RETURN
1065 IF b$<>"5" THEN LET p=w-s:
GO SUB d: GO TO VAL "1050"
1070 GO SUB g: PAUSE j: RETURN
1101 CLS : PLOT v-s,m+m: DRAW VA
L "128",a: PLOT v-s,VAL "103": D
RAW VAL "128",a: PLOT x,a: DRAW
o,m+m: DRAW a,q-s: DRAW -q,q: PL
OT VAL "240",a: DRAW -VAL "32",m
: DRAW a,VAL "119": DRAW u,u
1110 LET p=w-s: GO SUB f: GO SUB
 e: IF n>b*v THEN RETURN
1115 IF b$<>"7" THEN LET p=w: GO
 SUB d: GO TO 1100
1120 LET c=s: LET p=m+b: GO SUB
h
1151 CLS : PLOT v,m+m: DRAW VAL
"159",a: PLOT v,VAL "103": DRAW
VAL "159",a: PLOT x,a: DRAW o,VA
L "32": DRAW a,q-s: DRAW -q,q: P
LOT VAL "240",a: DRAW a,VAL "167
": DRAW l,l
1160 GO SUB f: GO SUB e: IF n>b*
v THEN RETURN
1165 IF b$<>"8" THEN LET p=w-s:
GO SUB d: GO TO VAL "1150"
1170 GO SUB g: PAUSE j: CLS : RE
TURN
1201 CLS : PLOT 15,0: DRAW 64,32
: FOR n=2 TO 8 STEP 2: PLOT 2*n+
80,n+32: NEXT n: PLOT 239,0: DRA
W -64,32: FOR n=2 TO 8 STEP 2: P
LOT -2*n+175,n+32: NEXT n: PLOT
8,175: DRAW 72,-72: FOR n=4 TO 1
6 STEP 4: PLOT 80+n,103-n: NEXT
n: PLOT 248,175: DRAW -72,-72::
FOR n=4 TO 16 STEP 4: PLOT 175-n
,103-n: NEXT n
1205 PRINT AT l+k,l+k;"CONTINUE"
;AT x-s,l+k;"FORWARDS"
1210 LET p=w-s: GO SUB f: GO SUB
 e: IF n>b*v THEN RETURN
1215 IF b$<>"7" THEN GO SUB d: G
O TO VAL "1201"
1230 LET p=m+b: LET c=a: GO SUB
h: RETURN
1301 CLS : PLOT 240,0: DRAW -32,
16: DRAW 0,119: DRAW 40,40: PLOT
 207,32: DRAW -32,0: DRAW 0,71:
DRAW 32,0: PLOT 175,32: DRAW -16
,8: FOR n=2 TO 8 STEP 2: PLOT -2
*n+159,n+39: NEXT n: PLOT 175,10
3: DRAW -16,-16: FOR n=4 TO 16 S
TEP 4: PLOT 159-n,87-n: NEXT n:
IF x$(i)="6" THEN GO TO 1402
1310 PLOT 15,0: DRAW 80,40: FOR
n=2 TO 8 STEP 2: PLOT 2*n+96,n+4
0: NEXT n: PLOT 8,175: DRAW 88,-
88: FOR n=4 TO 16 STEP 4: PLOT n
+96,87-n: NEXT n
1320 LET p=w-s: GO SUB f: GO SUB
 e: IF n>b*v THEN RETURN
1325 IF b$<>"7" THEN LET p=w: GO
 SUB d: GO TO VAL "1300"
1330 LET c=s: LET p=m+b: GO SUB
h
1350 CLS : PLOT 240,0: DRAW 0,16
7: DRAW 8,8: PLOT 239,32: DRAW -
64,0: DRAW 0,71: DRAW 64,0: PLOT
 175,32: DRAW -16,8: FOR n=2 TO
6 STEP 2: PLOT -2*n+159,n+40: NE
XT n: PLOT 175,103: DRAW -16,-16
: FOR n=4 TO 16 STEP 4: PLOT 159
-n,87-n: NEXT n: IF x$(i)="6" TH
EN GO TO 1451
1360 PLOT 15,0: DRAW 80,40: FOR
n=2 TO 8 STEP 2: PLOT 2*n+96,n+4
0: NEXT n: PLOT 8,175: DRAW 88,-
88: FOR n=4 TO 16 STEP 4: PLOT 9
6+n,87-n: NEXT n
1370 LET p=w-b: GO SUB f: GO SUB
 e: IF n>b*v THEN RETURN
1372 IF b$<>"8" AND b$<>"7" THEN
 GO SUB d: GO TO VAL "1350"
1375 IF b$<>"7" THEN GO SUB g: P
AUSE j
1378 IF b$<>"8" THEN LET p=m+s:
LET c=a: GO SUB h
1380 RETURN
1401 CLS
1402 PLOT 15,0: DRAW 32,16: DRAW
 0,119: DRAW -40,40: PLOT 48,32:
 DRAW 32,0: DRAW 0,71: DRAW -32,
0: PLOT 80,32: DRAW 16,8: FOR n=
2 TO 8 STEP 2: PLOT 2*n+96,n+40:
 NEXT n: PLOT 80,103: DRAW 16,-1
6: FOR n=4 TO 16 STEP 4: PLOT n+
96,87-n: NEXT n: IF x$(i)="6" TH
EN GO TO 1610
1410 PLOT 240,0: DRAW -80,40: FO
R n=2 TO 8 STEP 2: PLOT -2*n+159
,n+40: NEXT n: PLOT 248,175: DRA
W -88,-88: FOR n=4 TO 16 STEP 4:
 PLOT 159-n,87-n: NEXT n
1420 LET p=w-s: GO SUB f: GO SUB
 e: IF n>b*v THEN RETURN
1425 IF b$<>"7" THEN LET p=w: GO
 SUB d: GO TO VAL "1400"
1430 LET c=s: LET p=w-b: GO SUB
h
1450 CLS
1451 PLOT 15,0: DRAW 0,167: DRAW
 -8,8: PLOT 16,32: DRAW 64,0: DR
AW 0,71: DRAW -64,0: PLOT 80,32:
 DRAW 16,8: FOR n=2 TO 8 STEP 2:
 PLOT 2*n+96,n+40: NEXT n: PLOT
80,103: DRAW 16,-16: FOR n=4 TO
16 STEP 4: PLOT 96+n,87-n: NEXT
n: IF x$(i)="6" THEN GO TO 1660
1460 PLOT 239,0: DRAW -80,40: FO
R n=2 TO 8 STEP 2: PLOT -2*n+159
,n+40: NEXT n: PLOT 247,175: DRA
W -88,-88: FOR n=4 TO 16 STEP 4:
 PLOT 159-n,87-n: NEXT n
1470 GO SUB f: GO SUB e: IF n>b*
v THEN RETURN
1472 IF b$<>"5" AND b$<>"7" THEN
 LET p=w-s: GO SUB d: GO TO VAL
"1450"
1475 IF b$<>"7" THEN GO SUB g: P
AUSE j
1478 IF b$<>"5" THEN LET c=a: LE
T p=m+s: GO SUB h
1480 RETURN
1501 CLS : PLOT u+l,m+m: DRAW k*
u,a: PLOT u+l,VAL "103": DRAW k*
u,a: PLOT x,a: DRAW m+m,m: DRAW
a,VAL "119": DRAW -u,u: PLOT VAL
 "240",a: DRAW -VAL "32",m: DRAW
 a,VAL "119": DRAW u,u
1510 LET p=w-s: GO SUB f: GO SUB
 e: IF n>b*v THEN RETURN
1515 IF b$<>"7" THEN LET p=w: GO
 SUB d: GO TO VAL "1500"
1520 LET c=s: LET p=w-b: GO SUB
h
1551 CLS : PLOT m,m+m: DRAW VAL
"224",a: PLOT m,VAL "135": DRAW
VAL "224",a: PLOT x,a: DRAW a,VA
```

```
L "167": DRAW -l,l: PLOT VAL "24
0",a: DRAW a,VAL "167": DRAW l,l
1560 GO SUB f: GO SUB e: IF n>b*
v THEN RETURN
1565 IF b$<>"8" AND b$<>"5" THEN
 LET p=w-s: GO SUB d: GO TO VAL
"1550"
1570 GO SUB g: PAUSE j: RETURN
1601 GO TO VAL "1300"
1610 LET p=w-s: GO SUB f: GO SUB
 e: IF n>b*v THEN RETURN
1615 IF b$<>"7" THEN LET p=w: GO
 SUB d: GO TO VAL "1600"
1620 LET c=s: LET p=w-b: GO SUB
h
1650 GO TO VAL "1350"
1660 GO SUB f: GO SUB e: IF n>b*
v THEN RETURN
1665 IF b$<>"7" AND b$<>"8" AND
b$<>"5" THEN GO TO VAL "1650"
1670 IF b$="7" THEN LET p=m+s: G
O SUB h: RETURN
1680 IF b$="8" OR b$="5" THEN GO
 SUB g: PAUSE j: RETURN
1701 CLS : PLOT o,w+k: DRAW VAL
"127",a: PLOT o,VAL "119": DRAW
VAL "127",a: PLOT x,a: DRAW u+l,
w+k: DRAW a,v+x: DRAW -VAL "48",
u+l: PLOT VAL "240",a: DRAW -VAL
 "48",w+k: DRAW a,VAL "95": DRAW
 u+l,u+l: IF i>y THEN RETURN
1710 GO SUB f: PRINT AT l+b,l+t;
"DEAD END";AT m-b,x-k;"Back to t
he";AT m+s,x-s;"START": PAUSE b*
v: CLS : RETURN
2000 IF b$<>"7" THEN GO SUB VAL
"5000"
2001 CLS : PLOT a,m+m: DRAW VAL
"255",a: PLOT a,VAL "135": DRAW
VAL "255",a: PRINT AT l+b,l+t;"P
ASSAGE";AT m-k,m-b;"WALL"
2005 LET p=w-b: GO SUB f
2008 IF b$="7" THEN GO TO VAL "2
020"
2010 PRINT AT w,x-k;"TURN";AT w,
w-b;"BACK";AT w+s,l+t;"Press ";(
"8" AND b$="5")+("5" AND b$="8")
: PAUSE a: IF INKEY$<>"5" AND b$
="8" OR INKEY$<>"8" AND b$="5" T
HEN CLS : GO TO VAL "2001"
2015 PRINT AT w,x-k;"    ";AT w,
w-b;"    ";AT w+s,l+t;"     ":
 PAUSE b*x: PRINT AT w-s,m;"D";A
T w,m;("F" AND b$="8")+("E" AND
b$="5"): PAUSE b*x: CLS : RETURN

2020 PRINT AT w+s,l+t;"TRY AGAIN
": PAUSE b*x: CLS : RETURN
3000 PRINT AT w,b*t;"WHICH";AT w
,w-b;"WAY?";AT w+s,b*t;"Press";A
T w+s,w-b;"5,7 or 8": FOR n=s TO
 b*v: IF INKEY$<>"" THEN GO TO V
AL "3050"
3010 NEXT n
3020 PRINT AT x,l+k; INK k;"o";A
T m,l+k;"O";AT m+s,l+k;"B": PAUS
E b*x: PRINT AT k+l,x-k;"TOO SLO
W!";AT l+b,m-b;"The";AT x-k,l+b;
 INK k;"GREEN GOBLIN";AT m-k,m-t
; INK a;"sends you";AT x-b,l+b;"
BACK TO THE";AT m-b,l+k;" START
": PAUSE v+w: RETURN
3050 LET b$=INKEY$: IF b$="0" TH
EN STOP
3055 IF b$<>"7" AND b$<>"8" AND
b$<>"5" THEN GO TO 3000
3060 PRINT AT w,l+b;"     ";AT w
,w-b;"     ";AT w+s,l+b;"     ";A
T w+s,w-b;"     ": PAUSE w: R
ETURN
4001 PRINT AT p,m;"O";AT p+s,m;"
O";AT p+b,m;"B": RETURN
4600 LET p=w-s: GO SUB VAL "5000
": FOR n=(m AND b$="8")+(x AND b
$="5") TO (w+l AND b$="8")+(INT
PI AND b$="5") STEP (b$="8")-(b$
="5"): PRINT AT w-b,n;"GH";AT w-
s,n;"IJ";AT w,n;("KL" AND b$="8"
)+("MN" AND b$="5"): PAUSE l: PR
INT AT w-b,n;(" O" AND b$="8")+(
"O " AND b$="5");AT w-s,n;(" D"
AND b$="8")+("D " AND b$="5");AT
 w,n;(" E" AND b$="8")+("F " AND
 b$="5"): PAUSE l: NEXT n: PRINT
 AT w-b,n;("G" AND b$="8")+("H "
 AND b$="5");AT w-s,n;("I" AND b
$="8")+("J " AND b$="5");AT w,n;
("K" AND b$="8")+("N " AND b$="5
"): PAUSE l: PRINT AT w-b,n;" ";
AT w-s,n;" ";AT w,n;" ": RETURN
4800 PRINT AT p,m;"o";AT p+s,m;"
O";AT p+b,m;"P";AT p+INT PI,m;"
": PAUSE w: PRINT AT p-s,m;"Q";A
T p,m;"R";AT p+s,m;"S";AT p+b,m;
" ": PAUSE w
4810 IF c=s THEN LET c=a: RETURN

4820 PRINT AT p-s,m;"T";AT p,m;"
 ";AT p+s,m;" ": PAUSE w: LET c=
a: RETURN
5000 PRINT AT p,m;"D";AT p+s,m;(
"E" AND b$="8")+("F" AND b$="5")
: PAUSE b*x: RETURN
6000 BORDER 4: PRINT AT 1,5; INK
 4;"█████████████████████";AT 2,
5;"█                   ";AT 3,5
;"█████████████████████";AT 2,6;
 INK 0; PAPER 6;" LEPRECHAUN'S G
OLD ";AT 5,0; PAPER 7;"Can you m
ake you way to the    centre, a
nd find the GOLD?";AT 8,5;"Press
 5 to turn left";AT 9,11;"7 to m
ove forwards";AT 10,11;"8 to tur
n right";AT 11,10;"(0 to exit)";
AT 12,0;"Don't turn too early -
you'll   walk into the wall!";AT
 15,0; FLASH 1;"BEWARE!";AT 16,0
; FLASH 0;"The "; INK 4;"GREEN G
OBLIN"; INK 0;" will be        f
ollowing you. Don't take too   l
ong making a decision or he'll c
atch up and send you back to   t
he start"
6010 LET d=VAL "2000": LET e=VAL
 "3000": LET f=d+d: LET g=VAL "4
600": LET h=VAL "4800": LET j=VA
L "50": LET a=NOT d: LET s=SGN d
: LET b=s+s: LET k=b+b: LET l=k+
k: LET m=l+l: LET o=m*k: LET q=o
+l: LET t=k+s: LET u=t*l: LET v=
u+u: LET w=k*t: LET x=t*INT PI
6015 PRINT "PRESS ANY KEY TO BEG
IN": PAUSE 0: CLS : INPUT "Enter
 Difficulty Level (1 to 10)";z:
PRINT "PREPARING MAZE": LET y=z*
b+l: GO SUB 8000: GO TO 190
7000 GO SUB VAL "1700": PRINT AT
 m,x; INK VAL "6";"██";AT m+s,x-
b; INK k;"██████";AT m+b,x-b;"█
████";AT w-s,m-t; INK a;"O";AT w
,m-t;"C";AT w+s,m-t;"B";AT 16,21
; INK 4;"o";AT 17,21;"O";AT 18,2
1;"B"
7010 PRINT AT 9,17; INK 4; FLASH
 1;"CURSES!";AT 11,18; INK 0; FL
ASH 0;"You've";AT 12,19;"Found";
AT 13,21;"the";AT 14,20; INK 6;"
GOLD"
7020 PAUSE 100: PRINT AT 1,5;"PR
ESS 1 TO PLAY AGAIN": PAUSE 0: I
F INKEY$="1" THEN CLS : GO TO 60
15
7999 STOP
8000 LET x$="
              ": FOR i=s TO y STEP
b: LET x$(i)=STR$ INT (RND*INT P
I): LET x$(i+s)=STR$ INT (RND*k+
INT PI): IF i<INT PI THEN GO TO
VAL "8010"
8005 IF x$(i-s)<>"7" AND RND<VAL
 ".4" THEN LET x$(i+s)="7"
8010 NEXT i: RETURN
9000 LOAD "maze"CODE USR "a" : C
LS : GO TO 6000
```


THE FABULOUS CASSETTE
50
FROM cascade
ONLY £9.95 (INC.P&P and VAT)
VALUE that's out of this world
50 GAMES ON ONE CASSETTE
DRAGON BBC A/B Spectrum apple ATARI ORIC-1 ZX81 VIC 20

It is impossible to tell you everything about the 50 games on CASSETTE - 50 but they include many types such as maze, arcade, missile, tactical and logic games, to suit most tastes in computer game playing.
CASSETTE - 50 will appeal to people of all ages and the games will provide many hours of entertainment for all the family at a fraction of the cost of other computer games.

EXPRESS DELIVERY - ORDER NOW

Name ______________________

Address ______________________

____

__________ Post Code __________

Country ______________________

Dealers & Stockists enquiries welcome.

Please send me by return of post, Cassette 50 at £9.95 per tape. I enclose a cheque/postal order for

£ __________ made payable to Cascade Games Ltd.

Please debit my No. __________

SPECTRUM ☐ ORIC-1 ☐ ZX 81 ☐ VIC 20 ☐
BBC A/B ☐ DRAGON ☐ ATARI ☐ APPLE ☐ ☑

Cascade Games Ltd.,
Suite 4, 1-3 Haywra Crescent, Harrogate,
North Yorkshire, HG1 5BG. England.
Telephone: (0423) 504526.

PS 3/84

---

campbell systems

NEW! with MYRMIDONS Micro-Print 42/51 up to 51 columns!

Masterfile

Spectrum 48K

YES - Microdrive compatible!

Acclaimed as the definitive filing system for the 48K Spectrum – MASTERFILE'S machine coded flexibility gives you 32K (max) per file – 26 fields per record – up to 128 characters per field – multiple level searches for numeric or character comparisons – data presentation in any one of 36 user-defined displays which may be sequenced by any field – USER-BASIC for tailored processing. **" . . . the most comprehensive of the data-bases . . ."**

Sinclair User June 1983

With example file and detailed manual £15.00

Drawmaster

Spectrum 48K

DRAWMASTER is the ultimate DRAW AND PAINT utility for the Spectrum – compose your own HI-RES pictures with 8 direction 2 speed hand draw keys – fast fill – enlarge/reduce facility for all or part of the screen plus all the Spectrum colours and attributes – over 50 commands in all.

With detailed instructions £6.95

DLAN DISPLAY LANGUAGE

Spectrum 16/48K

DLAN ('dee-lan') DISPLAY LANGUAGE is a powerful machine coded interpreter for generating DYNAMIC TEXTUAL DISPLAYS for advertising and education – all commands are single letter or symbol – with 4-way scroll of 11 type-faces through user-defined windows.

With detailed manual £7.95

All programs mailed 1st class by return. Prices include VAT and postage within Europe. SAE for full list.

**Dept.(PSI) 15 Rous Road, Buckhurst Hill,
Essex, IG9 6BL England 01 504 0589**

CS

---

machine code sort in your Spectrum BASIC

ALL-SORT

simple to use

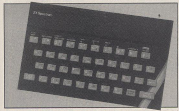
brilliant

£9.95

**Write for literature & stockists:
Alan Firminger
171 Herne Hill, London SE24**

CIRCUIT SKETCH

Turn on and tune in for a bit of electronic 'circuit training'.



This incredible program will allow Spectrum users to make full use of their machine's high resolution graphics to form diagrams of electronic apparatus and either SAVE the results on tape or PRINT them out on the ZX Printer. An example of the resolution possible with this listing is shown in Fig. 1, an outdated radio circuit.

DRAWING ON EXPERIENCE

When RUN, the user will be asked to type in the numbers corresponding to the INK and PAPER colours required. An initial point will also be asked for; this is the point from which the sketch is to begin from.

The drawing of the circuit can then begin. To draw the components, the key corresponding to the first letter of the component should be pressed. The following represents a list of components included in the program:

Non-polarised capacitor — 'c'
Variable capacitor — 'c'
Electrolytic capacitor — 'c'
Fixed resistor — 'r'
Variable resistor — 'r'
PNP transistor — 't'
Diode — 'd'
Inductor — 'i'
Switch — 's'
Battery — 'b'
Fuse — 'f'

Other components could easily be added should you require them.

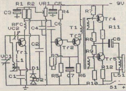
Once you have pressed a certain key, the computer requires other information as to the nature of the component. For example, information must be given as to whether it is to be drawn vertically or horizontally; whether it should be drawn from the positive or negative end; whether a resistor is variable or fixed, etc. Once a component has been drawn, you can draw the interlinking lines using the cursor keys, '5', '6', '7' and '8' to draw a line left, down, up or right respectively. The computer will automatically allow you to start drawing from the other side of the component, or in the case of a transistor, from the collector.

GOING UP . . .

Components may be drawn horizontally or vertically (except the transistor), but the following rules must be noted:

1 — Horizontal components are drawn left to right.
2 — Vertical components are drawn from the bottom to the top.
3 — If you are drawing from left to right, the component must be horizontal.

Fig.1. An example of the resolution available with this program. This circuit sketch is of an outdated radio circuit.



4 — Transistors are drawn horizontal, beginning at the base and ending up at the collector.

5 — Variable components are drawn from the central arrow position and terminate at the right or top end.

The current plotting position is always displayed; if you want to draw accurately or PRINT in component numbers or values at a later stage, you would be wise to note down the position of each component.

You may not draw off the screen; you will be stopped by the contents of line 64. However, you are allowed to change the plotting position by pressing the 'p' key and stating the co-ordinates of the new position. The variable screen is set to zero if further drawing will result in the line going off the screen.

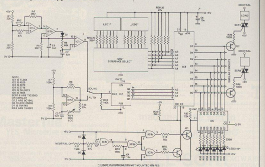
Once the circuit has been completed, you can press the Break key and, using PRINT, mark in any additional script onto the diagram. Should you wish to SAVE the program, try using the following:

SAVE "circuits" LINE 2

This will allow the program to RUN automatically when reLOADed.

Should the thought of playing around with electronic circuits scare you to death, try using it as a simple sketchpad.

A circuit diagram that may get a little involved!


+5V
R1 22k
R4 2M2
MIC
IC1a
IC1b
R3 47k
R2 22k
R5 2M7
C1 3u3
D1
R6 68k
C2 1u0
IC1c
0V
R16-23 330R
LED1*
LED2*
SW2* SEQUENCE SELECT
R30-35 22k
C6 10u
Vpp
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
IC4
D0
D1
D2
D3
D4
D5
D6
D7
OE CE
NEUTRAL
LOAD
SCR1
R39 100R
Q2
SCR8
R46 100R
SIMILAR CIRCUITS
NOTE
IC1 IS TL084
IC2 IS 4520
IC3 IS 4070
IC4 IS 2716
IC5 IS 74LS373
IC6 IS 7805
SCR1-8 ARE TIC206D OR SIMILAR
Q1,2 ARE BC184L
Q3-10 ARE 2N302
D1 IS 1N4148
D2-5 ARE 1N4001
R7 100k
R8 100k
IC1d
R9 100k
C3 1u0
RV1 500k LOG
R10 33k
SOUND
AUTO
R38 100k
EN
CLK IC2
RST
Q4
Q3
Q2
Q1
R36 22k
R37 22k
IC5
Q1
NEUTRAL
D2
D3
R11 150k
R12 47k
IC3a
IC3b
R13 150k
C4 10n
IC3c
IC3d
R14 15k
R15 15k
330R
LED3-10*
0V

* DENOTES COMPONENTS NOT MOUNTED ON PCB
CHECK MANUFACTURERS DATA FOR TYPE USED
FOR PIN CONNECTIONS

```
   1 REM CIRCUIT DIAGRAM SKETCH
  ZX SPECTRUM  © G.MAYNARD 1982
   2 INPUT "paper?";paper;"ink?"
;ink: INPUT AT 2,0;"initial plot
 position? x -";x;" Y- ";y
   3 GO SUB 8000
   4 PAPER paper: CLS : INK ink
   5 PLOT x,y
   6 LET screen=1
   8 PRINT AT 0,0;0;" , ";0
  10 LET m$=""
  15 LET m$=INKEY$
  30 IF m$>="5" AND m$<="8" THEN
 RESTORE (VAL m$-3)*100
  46 IF m$>="5" AND m$<="8" THEN
 GO TO 60
  50 RESTORE 55: FOR z=1 TO 10:
READ v$,v: IF v$=m$ THEN GO TO v
  52 NEXT z
  55 DATA "f",6500,"b",5500,"s",
6000,"i",3500,"p",1000,"",10,"c"
,2000,"r",2500,"t",3000,"d",4000
  60 READ a,b
  62 IF screen=0 THEN GO TO 64
  64 IF x=255 AND m$="8" OR x=0
AND m$="5" OR y=175 AND m$="7" O
R y=0 AND m$="6" THEN LET screen
=0: GO TO 10
  65 LET x=x+a
  67 LET y=y+b
  68 PRINT OVER 0;AT 0,0;"
"
  69 PRINT OVER 0;AT 0,0;x;" , "
;y
  70 DRAW a,b
  75 LET screen=1
  80 GO TO 10
 200 DATA -1,0
 300 DATA 0,-1
 400 DATA 0,1
 500 DATA 1,0
1000 INPUT "X - co-ordinate?";x
1005 INPUT "Y - co-ordinate";y
1010 PLOT x,y: GO TO 10
2005 INPUT "h or v";k$: GO SUB 8
```

```
000
2007 INPUT "electrolytic? (Y/N)"
;z$: GO SUB 8000: IF z$="y" THEN
 GO TO 2100
2010 INPUT "variable? (Y/N)";z$:
 GO SUB 8000: IF z$="y" THEN GO
TO 2050
2015 IF k$="h" THEN DRAW 0,8: DR
AW 0,-16: LET x=x+6: LET y=y-8:
PLOT x,y: DRAW 0,16: DRAW 0,-8:
LET y=y+8: GO TO 10
2020 IF k$="v" THEN DRAW -8,0: D
RAW 16,0: LET y=y+6: LET x=x+8:
PLOT x,y: DRAW -16,0: DRAW 8,0:
LET x=x-8: GO TO 10
2050 IF k$="h" THEN PLOT x-4,y-8
: DRAW 14,16: DRAW 0,-2: DRAW 0,
2: DRAW -2,0: DRAW 2,0: PLOT x,y
: GO TO 2015
2060 IF k$="v" THEN PLOT x-6,y-4
: DRAW 16,14: DRAW -2,0: DRAW 2,
0: DRAW 0,-2: DRAW 0,2: PLOT x,y
: GO TO 2020
2100 INPUT "pos or neg first?";z
$: GO SUB 8000
2110 IF k$="v" AND z$="pos" THEN
 DRAW 8,0: DRAW -16,0: DRAW 0,3:
 DRAW 16,0: DRAW 0,-3: PLOT x,y+
6: DRAW -8,0: DRAW 16,0: DRAW 0,
1: DRAW -16,0: DRAW 0,1: DRAW 16
,0: DRAW -8,0: LET y=y+9: GO TO
10
2120 IF k$="v" AND z$="neg" THEN
 DRAW 8,0: DRAW -16,0: DRAW 0,1:
 DRAW 16,0: DRAW 0,1: DRAW -16,0
: PLOT x,y+6: DRAW -8,0: DRAW 16
,0: DRAW 0,3: DRAW -16,0: DRAW 0
,-3: DRAW 0,3: DRAW 8,0: LET y=y
+9: GO TO 10
2130 IF k$="h" AND z$="pos" THEN
 DRAW 0,8: DRAW 0,-16: DRAW 3,0:
 DRAW 0,16: DRAW -3,0: PLOT x+6,
y: DRAW 0,8: DRAW 0,-16: DRAW 1,
0: DRAW 0,16: DRAW 1,0: DRAW 0,-
16: DRAW 0,8: LET x=x+9: GO TO 1
0
2140 IF k$="h" AND z$="neg" THEN
 DRAW 0,8: DRAW 0,-16: DRAW 1,0:
 DRAW 0,16: DRAW 1,0: DRAW 0,-16
: PLOT x+6,y: DRAW 0,8: DRAW 0,-
16: DRAW 3,0: DRAW 0,16: DRAW -3
,0: DRAW 3,0: DRAW 0,-8: LET x=x
+9: GO TO 10
2500 INPUT "h or v?";k$: GO SUB
8000
2505 INPUT "variable ? (Y/N)";z$
: GO SUB 8000
2506 IF z$="y" THEN GO TO 2700
2510 IF k$="h" THEN DRAW 0,4: DR
AW 18,0: DRAW 0,-8: DRAW -18,0:
DRAW 0,4: DRAW 0,-4: DRAW 18,0:
DRAW 0,4: LET x=x+18: GO TO 10
2520 IF k$="v" THEN DRAW 4,0: DR
AW 0,18: DRAW -8,0: DRAW 0,-18:
DRAW 4,0: DRAW -4,0: DRAW 0,18:
DRAW 4,0: LET y=y+18: GO TO 10
2700 IF k$="h" THEN DRAW -3,3: D
RAW 3,-3: DRAW -3,-3: DRAW 3,3:
DRAW 0,8: DRAW 0,-16: DRAW 6,0:
DRAW 0,16: DRAW -6,0: DRAW 3,0:
LET y=y+8: LET x=x+3: GO TO 10
2720 IF k$="v" THEN DRAW -3,-3:
DRAW 3,3: DRAW 3,-3: DRAW -3,3:
DRAW 8,0: DRAW -16,0: DRAW 0,6:
DRAW 16,0: DRAW 0,-6: DRAW 0,3:
LET x=x+8: LET y=y+3: GO TO 10
3000 INPUT "pnp or npn";z$: GO S
UB 8000
3010 IF z$="npn" THEN CIRCLE x+6
,y,8: PLOT x,y: DRAW 2,0: DRAW 0
,5: DRAW 0,-10: DRAW 1,0: DRAW 0
,10: DRAW 1,0: DRAW 0,-10: DRAW
0,3: DRAW 4,-4: DRAW 0,1: DRAW 0
,-1: DRAW -1,0: DRAW 1,0: DRAW -
4,4: DRAW 0,2: DRAW 6,4: LET x=x
+10: LET y=y+4: GO TO 10
3020 IF z$="pnp" THEN CIRCLE x+6
,y,8: PLOT x,y: DRAW 2,0: DRAW 0
,5: DRAW 0,-10: DRAW 1,0: DRAW 0
,10: DRAW 1,0: DRAW 0,-10: DRAW
0,3: DRAW 4,-4: DRAW -4,4: DRAW
1,0: DRAW -1,0: DRAW 0,-1: DRAW
0,1: DRAW 0,2: DRAW 6,4: LET x=x
+10: LET y=y+4: GO TO 10
3500 INPUT "h or v";z$: GO SUB 8
000
3510 IF z$="h" THEN DRAW 2,0: DR
AW 7,0,PI: DRAW -2,0: DRAW 7,0,P
I: DRAW -2,0: DRAW 7,0,PI: DRAW
2,0: LET x=x+21: GO TO 10
3520 IF z$="v" THEN DRAW 0,2: DR
AW 0,7,PI: DRAW 0,-2: DRAW 0,7,P
I: DRAW 0,-2: DRAW 0,7,PI: DRAW
0,2: LET y=y+21: GO TO 10
4000 INPUT "h or v?";z$: GO SUB
8000
4010 INPUT AT 0,0;"cathode (c) o
r anode (a) first?";k$: GO SUB 8
000
4020 IF z$="h" AND k$="c" THEN D
RAW 0,6: DRAW 0,-12: DRAW 0,6: D
RAW 7,-6: DRAW 0,12: DRAW -7,-6:
 DRAW 7,6: DRAW 0,-6: LET x=x+7:
 GO TO 10
4000 IF z$="h" AND k$="a" THEN D
RAW 0,6: DRAW 0,-12: DRAW 7,6: D
RAW -7,6: DRAW 7,-6: DRAW 0,6: D
RAW 0,-12: DRAW 0,6: LET x=x+7:
GO TO 10
4040 IF z$="v" AND k$="c" THEN D
RAW 6,0: DRAW -12,0: DRAW 6,0: D
RAW -6,7: DRAW 12,0: DRAW -6,-7:
 DRAW 6,7: DRAW -6,0: LET y=y+7:
 GO TO 10
4050 IF z$="v" AND k$="a" THEN D
RAW 6,0: DRAW -12,0: DRAW 6,7: D
RAW 6,-7: DRAW -6,7: DRAW -6,0:
DRAW 12,0: DRAW -6,0: LET y=y+7:
 GO TO 10
5500 INPUT "h or v";z$: GO SUB 8
000
5510 INPUT "pos or neg first";k$
: GO SUB 8000
5520 IF z$="v" AND k$="pos" THEN
 DRAW -7,0: DRAW 14,0: PLOT x,y+
3: DRAW 3,0: DRAW -6,0: DRAW 3,0
: LET y=y+3: GO TO 10
5530 IF z$="v" AND k$="neg" THEN
 DRAW 3,0: DRAW -6,0: PLOT x,y+3
: DRAW -7,0: DRAW 14,0: DRAW -7,
0: LET y=y+3: GO TO 10
5540 IF z$="h" AND k$="pos" THEN
 DRAW 0,7: DRAW 0,-14: PLOT x+3,
y: DRAW 0,3: DRAW 0,-6: DRAW 0,3
: LET x=x+3: GO TO 10
5550 IF z$="h" AND k$="neg" THEN
 DRAW 0,3: DRAW 0,-6: PLOT x+3,y
: DRAW 0,-7: DRAW 0,14: DRAW 0,-
7: LET x=x+3: GO TO 10
6000 INPUT "h or v?";k$: GO SUB
8000
6010 IF k$="h" THEN DRAW 14,7: L
ET x=x+14: PLOT x,y: GO TO 10
6020 IF k$="v" THEN DRAW 7,14: L
ET y=y+14: PLOT x,y: GO TO 10
6500 INPUT "h orv?";k$: GO SUB 8
000
6510 IF k$="h" THEN DRAW 0,4: DR
AW 0,-8: DRAW 1,0: DRAW 0,8: DRA
W 1,0: DRAW 0,-8: DRAW 1,0: DRAW
 0,8: DRAW 10,0: DRAW 0,-8: DRAW
 -10,0: DRAW 10,0: DRAW 0,4: LET
 x=x+13: GO TO 10
6520 IF k$="v" THEN DRAW 4,0: DR
AW -8,0: DRAW 0,1: DRAW 8,0: DRA
W 0,1: DRAW -8,0: DRAW 0,1: DRAW
 8,0: DRAW 0,10: DRAW -8,0: DRAW
 0,-10: DRAW 0,10: DRAW 4,0: LET
 y=y+13: GO TO 10
8000 FOR z=1 TO 7: NEXT z: RETUR
N
```

Colin Gooch

BATS

Let this 16K program drive you batty!

In this program, you are a bat flying about trying to secure some moths for your supper. Movement is made via the four 'arrow' keys on the Spectrum, and to make a catch you have to place the centre of the bat character directly over the moth. To make it just that little bit more difficult, the moth moves totally at random.

Because your bat cannot fly in bright light, the moth can escape you by flying into the lighted window in the top left-hand corner of the screen. All you have to do is to trap the moth before it gets there! The game ends when you have attempted to catch ten moths, although there is a bonus if you manage to catch all ten.

HOLY BAT-CATCHER!

Initialisation of the graphics is done in lines 2230 to 2250. This leads straight into the instruction page which is completed by lines 2300 to 3030. This last section is not repeated when you play subsequent games after your first. The main program begins with line 50 which calls the subroutine to print the screen display, followed by lines 90 to 120 which initialise the screen co-ordinates for the bat and the moth. Lines 200 to 300 are a loop which constantly moves the position of the moth, and if the right keys are pressed this section of the program actually moves the bat.

The conditional statements for the moth make sure that it stays within the screen area and also reverse its direction should it get too near to the edge. The moth is moved 'dx' and 'dy' across and up the screen, but it is not incremented on each pass of the loop. It is the RND value in lines 230 and 240 which look after this latter task, and it is this which gives the moth its random movement.

The conditional statements for the bat read the keyboard, keep the bat on the screen and also ensure that it will not move into the area of the window. Line 320 detects a caught moth and moves on to the scoring section of the program. Lines 330 to 350 detect the moth in the vicinity of the window, and once there it is automatically drawn into the window, the moth's home score is updated and a new moth appears down in the bottom right-hand area of the screen.

Scoring is carried out from lines 370 to 520. This includes a 'high score' routine, and offers you a bonus 'go' if you managed to catch ten moths out of ten. This invitation comes courtesy of lines 530 to 580.

FLYING TONIGHT

The REM at line 20 indicates which lines need letters typed in the graphics mode. Lines 250 and 260 use keyboard graphics and it is important that you get four blanks at the end of each string.

Line 3400 looks the most complicated. However, if you decide that you don't want to put a title or instructions in your listing, you could always rewrite line 2400 to read:

```
2400  RETURN
```

and omit all from thereon.

```
   9 REM ************************
  10 REM BATS by C N GOOCH
  11 REM ************************
  20 REM In lines 250,300,570,23
00 the letters to be printed are
 those letters in GRAPHICS mode
  30 RANDOMIZE : CLS : BORDER 1:
 LET HSC=0: GO SUB 2200
  40 PAPER 5: CLS : LET TET=0: L
ET ET=0: LET MO=0
  50 GO SUB 2050
  90 LET xx=16: LET yy=20
 100 LET y=12+INT (RND*5): LET x
=0: IF RND>.4 THEN LET X=31
 120 LET dX=1: LET dy=1
 190 REM MOVE MOTH
 200 LET dX=((X=0)-(X=31)+dX*(X>
0 AND x<31))
 210 LET dy=(y=0)-(y=20)+dy*(y>0
 AND y<20)
 220 PRINT AT y,x; PAPER '5+1*(X<
8 AND Y<8);" "
 230 IF RND>.6 THEN LET x=x+dX
 240 IF RND>.5 THEN LET y=y+dy
 250 PRINT AT y,x; PAPER 5+1*(X<
7 AND Y<7);"A"
 260 REM MOVE BAT
 270 PRINT AT yy,xx;"   "
 280 LET yy=yy+((INKEY$="6" AND
yy<20)-(INKEY$="7" AND yy>0))*(Y
Y>8 OR XX>8 OR INKEY$="6")
 290 LET xx=xx+((INKEY$="8" AND
xx<29)-(INKEY$="5" AND xx>0))*(Y
Y>8 OR XX>8 OR INKEY$="8")
 300 PRINT AT yy,xx;"BCD"
 310 REM MOTH CATCH OR HOME
 320 IF X=XX+1 AND y=yy THEN GO
TO 370
 330 IF Y<8 AND X<8 THEN LET DX=
-1: LET DY=-1
 340 IF X<4 AND Y<4 THEN GO SUB
2080: LET MO=MO+1: PRINT AT 1,1;
 PAPER 7;MO: IF MO=10 THEN GO TO
 410
 350 IF X<4 AND Y<4 THEN GO TO 1
00
 360 GO TO 200
 370 REM MOTH CAUGHT
 380 IF INKEY$<>"" THEN GO TO 38
0
 390 LET ET=ET+1: PRINT AT 21,5;
 INK 7; PAPER 2; BRIGHT 1;" MOTH
S CAUGHT ";ET: PAUSE 100: IF ET>
=10 THEN GO TO 450
 400 GO TO 100
 410 REM **END ROUND**
 420 LET TET=TET+ET: IF ET=10 AN
D MO=0 THEN PRINT AT 15,0; PAPER
 4;" IT'S A GOOD NIGHT FOR HUNTI
NG  ";AT 17,10; PAPER 2; INK 7;
BRIGHT 1; FLASH 1;" CARRY ON ":
IF INKEY$<>"" THEN GO TO 420
 430 IF ET=10 AND MO=0 THEN PAUS
E 400: FOR N=15 TO 17 STEP 2: FO
R M=0 TO 31: PRINT AT N,M; PAPER
 5;" ": NEXT M: NEXT N: GO TO 10
0
 480 IF TET>HSC THEN LET HSC=TET
 490 PAPER 4: CLS : PRINT AT 5,0
; PAPER 6;"  YOUR SCORE THAT ROU
ND WAS     "'" ";ET;" MOTHS EATE
N ";MO;" MOTHS ESCAPED "
 500 IF TET>10 THEN PRINT AT 7,0
; PAPER 6;" YOUR TOTAL SCORE WAS
 ";TET
 510 IF ET>HSC THEN LET HSC=ET
 520 PRINT AT 10,6; PAPER 3; INK
 0;" HIGHEST SCORE SO FAR ";HSC
 530 PRINT AT 12,4; PAPER 6;" DO
 YOU WANT ANOTHER GO ?";AT 13,4;
" PRESS ""Y""es TO CONTINUE ";AT
 14,4;" ""N""o TO STOP OR ""C""h
ange";AT 15,4;" TO GO TO NEXT GA
ME      "
 540 IF INKEY$="Y" OR INKEY$="y"
 THEN CLS : GO TO 40
 550 IF INKEY$="N" OR INKEY$="n"
 THEN GO TO 570
 555 IF INKEY$="C" OR INKEY$="c"
 THEN CLS : PRINT AT  10,10; FLA
SH 1; PAPER 6;" START THE TAPE "
: LOAD ""
 560 GO TO 540
 570 CLS : FOR N=0 TO 21 STEP 2:
 FOR M=0 TO 31 STEP 4: PRINT AT
N,M;"BCD": NEXT M: NEXT N
 580 PRINT AT 10,4; INK 7; PAPER
 1; BRIGHT 1; FLASH 1;"  BYE BY
E  ";AT 14,12; PAPER 2; INVERSE
1;" MIND THE VAMPIRES ": GO TO 5
80
2030 REM WINDOW
2050 LET A$="▛▀▜     "
2060 LET B$="▌ ▐     "
2070 FOR N=0 TO 7: FOR M=0 TO 7:
 PRINT AT N,M; PAPER 6;" "
2080 NEXT M: NEXT N: FOR N=0 TO
3: PRINT AT N,0; PAPER 6;B$: NEX
T N: FOR N=0 TO 3 STEP 3: PRINT
AT N,0; PAPER 6;A$: NEXT N
2090 RETURN
2200 REM ***MOTH***
2210 RESTORE 2210: DATA 0,36,102
,255,126,102,36,36
2220 REM >>>BAT<<<
2230 DATA 8,28,62,63,127,231,195
,128,36,60,60,24,255,255,255,126
,16,56,124,252,254,231,195,1
2240 REM ###FORM GRAPHICS###
2250 FOR N=0 TO 3: FOR M=0 TO 7:
 READ A: POKE USR CHR$ (N+CODE "
A")+M,A: NEXT M: NEXT N
2300 FOR N=0 TO 21 STEP 2: FOR M
=1 TO 31 STEP 4: PRINT AT N,M; I
NK 1+INT (RND*3);"BCD": NEXT M:
NEXT N
2400 GO SUB 3400
2500 PRINT AT 6,0;K$
2600 PRINT AT 14,2; PAPER 6;" PR
ESS ""ENTER"" TO CONTINUE ";AT 2
1,10; PAPER 7; INK 0; FLASH 1;"
STOP THE TAPE ": PAUSE 0: CLS
2700 PRINT AT 2,10; PAPER 2; INK
 7; BRIGHT 1; FLASH 1;" B A T S
 "
2800 PRINT AT 4,0;" YOU ARE A BA
T HUNTING MOTHS      "'"FOR SUPPER
 ."
2900 PRINT AT 6,0;" THE BAT IS M
OVED WITH KEYS No.  "'" ""5 TO 8"
" "
3000 PRINT AT 8,0;" THE MOTH IS
CAUGHT WHEN THE  "'"BAT IS CENTRA
LLY OVER IT"
3010 PRINT AT 10,0;" THE MOTH IS
 SAFE IN THE YELLOW "'"AREA OF T
HE LIGHTED WINDOW"'" THE BAT CAN
NOT GO HERE"
3020 PRINT AT 16,0;" THE GAME EN
DS WITH 10 MOTHS"'" ""HOME"" OR
""CAUGHT"""
3030 PRINT AT 18,0;" PRESS ""ENT
ER"" TO CONTINUE ": PAUSE 0: CLS
 : RETURN
3400 LET K$="
"S      BAT
         "
3410 RETURN
```

LOTUS-SOFT

any spectrum
LORD HARRY
&
LADY HARRIET

Kempston-type joystick or keyboard
100% machine code. Special animated cartoon graphics. Unique change screens features. Avoid the vicious dogs, find the hidden exit and escape. (5 star review).

ZX SPECTRUM
16K or 48K
HAWKS

SHOOT EM UP
Eggs fall on parachutes, watch them hatch Mutant Birds. Fiendish Hawks, Laser, Missiles. 100% machine code. Progressive levels.

48K spectrum
Bowls

Graphics simulation of the popular game of bowls. Watch the ball swerve towards the jack. But have you chosen the weight and angle correctly?

BULL TABLES
also nine lives 'candyman'
5 years to Genius
any spectrum

Fun-time educational game. Full feature arithmetic. Do your sums quickly or the bull will escape. Also—Nine Lives Candyman.

ALL CASSETTES AT ONLY £5:50 EACH, INCLUDING FIRST CLASS POST & PACKING

Send cheque/postal order to:
LOTUS-SOFT, 43 MAES ROAD, LLANGENNECH, LLANELLI, DYFED SA14 8UH

Name ....
Address ....
....
....
(BLOCK CAPITALS PLEASE)

LORD HARRY ☐ Please
HAWKS ☐ Tick
BOWLS ☐ ✓
BULLTABLES ☐

Total Enclosed £

See our games and much more at
POSI-TRON COMPUTING
3 STEPNEY ARCADE
LLANELLI (05542) 59624

And at REGAL SOFTWARE, GILLINGHAM
GAMES CENTRE, LONDON
and other good retail outlets

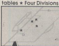
TRADE ENQUIRIES WELCOME

The No.1* Football Game

ZX81 16K
ZX Spectrum 48K

Football Manager
Designed by Kevin Toms

Some of the features of the game:
★ Matches in 3D graphics ★ Transfer market ★ Promotion and relegation ★ F.A. Cup matches ★ Injury problems ★ Full league tables ★ Four Divisions



★ Pick your own team for each match. ★ As many seasons as you like ★ Managerial rating ★ 7 skill levels ★ Save game facility.

Addictive

* ZX81 Chart
Home Computing Weekly
1.8.83 and 1.11.83.

Comments about the game from press and our customers.
"FOOTBALL MANAGER is the best game I have yet seen on the Spectrum and my personal favourite of all the games on any micro .... To the ordinary person it is an excellent view of what can be done in the field of computer games .... The crowning glory of this game is the short set pieces of match highlights which show little stick men running around a pitch, shooting, defending and scoring .... It is a compulsive game but people who cannot take game sessions of 9 hours or so, which happened on one happy Sunday, will be grateful to know that there is a 'save to tape' option. FOOTBALL MANAGER has everything it could .... The originator, Addictive Games, certainly deserve the name." Rating: 19/20 (Practical Computing – August 1983).

From software stockists nationwide, inc WHSMITH John Menzies

Prices: Spectrum 48K £6.95
ZX81 16K £5.95
(N.B. 3D GRAPHICS ARE NOT INCLUDED IN THE ZX81 VERSION)
Overseas orders add £1.50

To order by mail (p&p free) send cheques or postal orders to
Addictive Games
7A RICHMOND HILL
BOURNEMOUTH BH2 6HE

Dealers! For urgent stocks send your headed notepaper direct to our address.

ZX81 – FORTH ROM

with multi-tasking

Runs more than 10 tasks at once. Schedule tasks to run from 50 times per second to once a year. Ideal for control purposes. Three times faster than fig-FORTH, but fig compatible. Available as a *'fit-it-yourself'* EPROM, with an extensive manual for £25 plus VAT. Some ready-converted ZX81's available.

Coming Soon!

FORTH-I/O cartridge for Spectrum £59 plus VAT.
Ask for details

David Husband
2 Gorleston Road, Branksome,
Poole BH12 1NW
Telephone: 0202 302385

3D NOUGHTS AND CROSSES

Add a new dimension to this old game in 48K.

This is a game played on a cubical 'board' containing 27 spaces, any one of which can be filled with a nought or a cross. The board is represented on the screen by three conventional noughts and crosses boards placed one above the other. The levels are labelled A, B and C and the spaces on each level are numbered from one to nine. The object of the game is not to make the first row of three – that is too easy for the player who goes first – but to make more complete rows than your opponent, the computer.

When the program is RUN, the computer asks you to choose a skill level, then draws the board and asks if you want to go first. The computer plays the 'X' and you play the 'O'. Any row of three will count; some of them are hard to spot at first, but you'll soon get the hang of it. There are 49 possible rows. Examples: A1, B1 and C1 make a row, and so ▶

do A1, B5 and C9. If you change your mind halfway through entering a move, press any obviously wrong key and you will be able to start your entry again.

In between each move the computer checks through all the possible rows and counts any completed ones by making an audible blip. The blip is higher for the computer's rows than for yours, so you can listen to your progress.

The computer is hard to beat at first, especially at level three, but it can be done, especially if you go first! Why not try to improve the computer's game? You can see how its strategy is organised by reading the program notes and you could, for example, change the order in which it checks for almost complete rows by changing the order of the numbers 1-49 in the DATA statements at lines 1060-1080.

VARIABLES

To save on memory use, most of the variables are only temporary, and are used for different purposes in different parts of the program. These are the variables which have a single constant use:

a$ — Contains the character of the key last pressed when an input was requested.
a() — An array containing one element for each of the 27 spaces on the board.
b() — An array containing one element for each of the 49 possible rows of three.
level — The skill level which has been selected.
posn — The number of the element of a() corresponding to the move which has been selected by you or the computer.
go — The number of moves which have been made.
m — An indicator of whose turn it is to move.

Other variables used are: g, f, a, b, p, q, t and no.

MEMORY

As listed, this program will not run on a 16K machine — it uses more than 8,300 bytes. However, it could be compressed for a 16K machine by omitting the REM statements and perhaps using some of the standard Sinclair memory saving methods. For example, numbers can be referred to using the VAL function, eg, PRINT AT VAL"22",VAL"2"; and so on. Again, some of the lines can be run together with the use of colons.

PROGRAM NOTES

Line 5 Calls the instructions subroutine at line 9900.

Line 20-70 Define the 'X' and 'O' graphics characters. This short routine, with the appropriate DATA statements, can be used to define any number of characters up to the maximum of 21.

Line 80-90 Initialise the arrays a() and b(), and the move counter, and set the screen attributes for the game.

Line 95 POKE 23658,8 sets Caps Lock at capitals, which simplifies mug-trapping. POKE 23659,1 allows you to PRINT AT line 22.

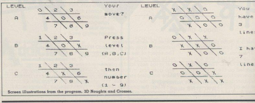
Line 100-175 Set up the board.

Lines 180-198 Set who is to go first. Line 190 calls an INKEY$ subroutine at line 9500 which stores the key pressed in a$. Note line 194; the effect of the commas after a PRINT AT statement is to blank out half a line per comma in the current PAPER colour. m is set to zero when it is the computer's move, and one when it is your move. If the computer is to go first, the program jumps to 1000.

Lines 200-350 Ask for your move, and mug-trap the results (again returned in a$). An acceptable move is turned into a number from one to 27 and stored in the variable, posn, for use later when displaying the move on the screen. The corresponding element of array a() is checked to see if it is empty; if it is, a four is inserted. (Four is used for human's moves, and one for the computer's moves.) The program then jumps to line 8000.

Lines 1000-7910 These are used for the computer's moves.

Line 1005 Lets you know that the computer is working out its move. After each stage of



Your move?

Press level (A,B,C) then number (1 - 9)

You have 3 lines: I have 7 lines.

Screen illustrations from the program, 3D Noughts and Crosses.

working, if the computer has not found an acceptable move it will add an extra full stop after 'I'M THINKING . . .' so that you can see how hard it has had to think!

Lines 1010-1025   If the centre space is empty, the computer will always choose to go there. The centre space corresponds to element 14 in array a(). Once a move has been chosen, the computer always jumps to line 8000. If the centre space has already been filled, the computer will move on to line 1030.

Line 1030   At skill level one, the computer skips the rest of its strategy and jumps to a routine at line 7000 which generates a random move.

Lines 1060-1080   Contain DATA which tells the computer the order in which it is to scan array b() for the next steps in its strategy.

Line 1105   At skill level two, the computer skips the next two steps of its strategy.

Lines 1110-1135   Each element of array b() contains the sum of the numbers held in three elements of array a() corresponding to a row of three spaces on the board. For example, a(1) plus a(2) plus a(3) corresponds to the state of the row form the top far left of the board (space A1) to the top far right (space A3). This total is stored in one of the elements of array b(). In this section of the program, the computer checks through array b() looking for the number '8', which represents a row on the board containing two 'O's and an empty space. If the computer finds such a row, it jumps to a routine at line 7500 which will identify the empty space so that the computer can put an 'X' there to block your row.

Lines 1140-1175   If you are not in a position to complete a row next move, the computer checks through array b() again, this time looking for the number '2'. This corresponds to a row on the board containing two Xs and a blank space. If such a row is found, the computer again jumps to line 7500 so that it can identify the empty space and complete a row of Xs.

Lines 1180-2010   Failing these, the computer looks for a row containing two blank spaces and one X. If it finds one, it will inset another X, threatening to complete a row on its next move.

Lines 2015-2055   If the computer can find none of these possibilities, it will check array a() in the order given in the DATA statement in line 2055 to see of any of the centre face or corner spaces are empty. It will make its move in the first one it finds. Plenty of space is left between lines 2055 and 7000 to insert more checks if you want to elaborate on the computer's strategy.

Lines 7000-7030   Generate a random move in any empty space on the board.

Lines 7500-7800   Contain a routine used by the computer to find which of the spaces, in a line it has identified in one of its scans of array b(), is empty. On exiting from this routine, the temporary variable, a indicates the appropriate empty element in array a().

Lines 7900   Puts the comuter's move into array a() and also into the variable, posn, for later use when displaying the move.

Lines 8000-8080   Find the correct screen address for the selected move and print an 'X' or 'O', as appropriate. The 'O' is a Graphics A and the 'X' is a Graphics B. The selected move is also displayed at the bottom of the screen.

Lines 8500-8745   Scan through array a() in all the combinations of three elements which correspond to a row of three on the board. In each case the sum of the three elements is stored in an element of array b() by means of a subroutine at line 9600. The temporary variable, a, is given the value 9600 to shorten the repeated calls to this subroutine.

Lines 8750-8770   Increase the move counter by one and check for the end of the game.

Line 8810   Blanks out the message line at the bottom of the screen.

Lines 8220   If m is one, it becomes zero; if zero, it becomes one. This indicates whether it is your move or the computer's. This neat line was borrowed from J.A. Enness's program, Squareology, in ***ZX Computing***, Summer 1982.

Line 8830   Jumps to line 1000 if it is the computer's move next, or line 200 if it is your move next.

Lines 9000-9050   When the game is over, this routine counts the completed rows of Os and Xs by scanning array b().

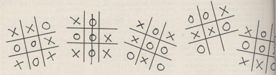
Lines 9200-9480   Display the final score and suitable messages, with jubilant or otherwise sound effects. (Room for improvement here, if you're interested.) You are asked if you want to play again. After the POKEs in line 95, if you break into direct command mode without POKE 23659,2 (line 9460) you will get a ***very*** strange screen display!

Lines 9500-9540   Contain an INKEY$ subroutine which waits for a key to be pressed, and stores the result in a $ for return.

Lines 9600-9640   Contain the subroutine used by the section of the program starting at line 8500. After each move, it stores the state of each possible row of three in an element of array b(). It gives a low blip when it finds a row of Os, and a higher one when it finds a row of Xs. It keeps quite about mixed or empty lines.

Lines 9900-9995   Contain the subroutine which displays the instructions and sets the skill level.

►



```
   1 REM 3-D NOUGHTS AND CROSSES
   2 REM
   3 REM by Bill Welch, Mar 1983
   4 REM
   5 GO SUB 9900
  10 REM DEFINE CHARACTERS
  20 FOR g=144 TO 145: FOR f=0 T
O 7
  30 READ a
  40 POKE USR CHR$ g+f,a
  50 NEXT f: NEXT g
  60 DATA 0,120,132,130,65,33,30
,0
  70 DATA 0,136,72,56,24,20,18,1
7
  80 REM INITIALISE
  85 DIM a(27): DIM b(49): LET g
o=0
  90 BORDER 1: PAPER 1: INK 4: C
LS
  95 POKE 23658,8: POKE 23659,1
 100 REM PRINT BOARD
 105 FOR f=1 TO 15 STEP 7
 110 LET b=8: LET a=49
 115 FOR p=f TO f+5 STEP 2
 120 FOR q=b TO b+9 STEP 4
 125 PRINT AT p,q; INK 5;CHR$ a
 130 LET a=a+1: NEXT q
 135 LET b=b+2
 140 NEXT p: NEXT f
 145 FOR f=14 TO 127 STEP 56
 150 PLOT 122,f: DRAW -44,44
 155 PLOT 154,f: DRAW -44,44
 160 PLOT 80,f+14: DRAW 84,0
 165 PLOT 64,f+30: DRAW 84,0
 170 NEXT f
 175 INK 6: PRINT AT 0,0;"LEVEL"
;AT 3,2;"A";AT 10,2;"B";AT 17,2;
"C"
 180 REM DECIDE WHO GOES FIRST
 185 PRINT AT 22,2;"Do you want
to go first?": BEEP .05,40
 190 GO SUB 9500
 192 IF a$<>"Y" AND a$<>"N" THEN
 BEEP .1,-24: GO TO 190
 194 PRINT AT 22,2,,
 196 IF a$="N" THEN LET m=0: GO
TO 1000
 198 LET m=1
 200 REM HUMAN'S MOVE
 210 PRINT AT 0,25;"Your";AT 2,2
5;"move?"
 220 PRINT AT 8,25;"Press";AT 10
,25;"level";AT 12,24;"(A,B,C)"
 230 PRINT AT 16,25;"then";AT 18
,25;"number";AT 20,24;"(1 - 9)"
 235 BEEP .05,40
 240 GO SUB 9500
 245 IF a$<>"A" AND a$<>"B" AND
a$<>"C" THEN BEEP .1,-24: GO TO
240
 250 BEEP .08,31
 260 LET a=CODE a$-64
 270 PRINT AT 22,0;"YOUR MOVE IS
 ";a$;"-              ";: PRINT
 AT 22,15;
 280 GO SUB 9500
 290 IF a$<"1" OR a$>"9" THEN BE
EP .1,-24: PRINT "? Start again.
": GO TO 240
```

```
 300 PRINT a$: BEEP .1,31
 310 LET posn=VAL a$+(9 AND a=2)
+(18 AND a=3)
 320 IF a(posn)<>0 THEN BEEP .1,
-24: PRINT AT 22,0;"That place i
s full: try again.": GO TO 240
 330 LET a(posn)=4
 340 FOR p=0 TO 21 STEP 2: PRINT
 AT p,24;"      ": NEXT p
 350 GO TO 8000
1000 REM COMPUTER'S MOVE
1005 PRINT AT 22,0;"I'M THINKING
...";: BEEP .005,12
1010 REM a) centre first?
1020 IF a(14)=0 THEN LET a(14)=1
: LET posn=14: GO TO 8000
1025 PRINT ".";: BEEP .005,12
1030 IF level=1 THEN GO TO 7000
1050 REM [illegible]
1060 DATA 14,23,5,48,46,49,47,33
,28,39,36,45,43,40,37,30,31,34,4
2
1070 DATA 25,1,12,9,10,19,7,3,18
,21,27,16,35,44,32,29,41,38
1080 DATA 11,20,15,17,2,8,4,26,2
2,6,13,24
1100 REM b) first, human's lines
1105 IF level=2 THEN GO TO 1180
1110 RESTORE 1060
1115 FOR f=1 TO 49
1120 READ t
1125 IF b(t)=8 THEN GO TO 7500
1130 NEXT f
1135 PRINT ".";: BEEP .005,12
1140 REM c) computer's lines
1145 RESTORE 1060
1150 FOR f=1 TO 49
1155 READ t
1160 IF b(t)=2 THEN GO TO 7500
1170 NEXT f
1175 PRINT ".";: BEEP .005,12
1180 REM d) computer line 2 up?
1185 RESTORE 1060
1190 FOR f=1 TO 49
1195 READ t
2000 IF b(t)=1 THEN GO TO 7500
2005 NEXT f
2010 PRINT ".";: BEEP .005,12
2015 REM If not, look for empty
     [illegible] and corners.
2020 RESTORE 2055
2025 FOR f=1 TO 14
2030 READ t
2035 IF a(t)=0 THEN LET posn=t:
LET a(t)=1: GO TO 8000
2040 NEXT f
2050 PRINT ".";: BEEP .005,12
2055 DATA 15,23,5,13,23,11,27,19
,3,7,1,21,25,9
7000 REM If none, random move.
7010 LET posn=INT (RND*27)+1
7020 IF a(posn)=0 THEN LET a(pos
n)=1: GO TO 8000
7030 BEEP .005,12: GO TO 7000
7500 REM [illegible] line.
7505 LET a=0
7510 IF t<10 THEN FOR p=t TO t+1
8 STEP 9
7515 IF t>9 AND t<19 THEN LET q=
((t-9) AND t<13)+((t-3) AND (t<1
6 AND t>12))+((t+3) AND t>15): F
OR p=q TO q+7 STEP 3
7520 IF t>18 AND t<23 THEN LET q
=(t-19)*3+1: FOR p=q TO q+2
7530 IF t>27 AND t<31 THEN LET q
=t-27: FOR p=q TO q+24 STEP 12
7540 IF t>30 AND t<34 THEN LET q
=(t-31)*3+1: FOR p=q TO q+20 STE
P 10
7550 IF t>33 AND t<37 THEN LET q
=t-27: FOR p=q TO q+12 STEP 6
7560 IF t>36 AND t<40 THEN LET q
=(t-37)*3+3: FOR p=q TO q+16 STE
P 8
7570 IF t>39 AND t<43 THEN LET q
=(t-40)*9+1: FOR p=q TO q+8 STEP
 4
7580 IF t>42 AND t<46 THEN LET q
=(t-43)*9+3: FOR p=q TO q+4 STEP
 2
7590 IF t=46 THEN FOR p=1 TO 27
STEP 13
7600 IF t=47 THEN FOR p=3 TO 25
STEP 11
7610 IF t=48 THEN FOR p=7 TO 21
STEP 7
7620 IF t=49 THEN FOR p=9 TO 19
STEP 5
7800 LET a=a+(p AND (a=0 AND a(p
)=0)): NEXT p
7900 REM Record computer's move
7910 LET posn=a: LET a(a)=1
8000 REM PRINT MOVE
8010 IF m=0 THEN PRINT " ";CHR$
(CODE "B"-(1 AND posn<10)+(1 AND
 posn>18));"-";posn-(9 AND posn>
9)-(9 AND posn>18)
8030 LET a=8-(7 AND posn<10)+(7
AND posn>18)
8040 LET no=posn-(18 AND a=15)-(
9 AND a=8)
8050 LET p=a+(2 AND no>3)+(2 AND
 no>6)
8060 LET q=p-a+8
8070 LET q=q+(4 AND (no=2 OR no=
5 OR no=8))+(8 AND (no=3 OR no=6
 OR no=9))
8080 BEEP .01,36: PRINT INK 7; B
RIGHT 1;AT p,q;("A" AND m=1)+("B
" AND m=0)
8500 REM SEARCH THROUGH LINES
8505 LET a=9600: LET b=1
8510 FOR p=1 TO 9
8515 LET q=a(p)+a(p+9)+a(p+18)
8520 GO SUB a
8525 NEXT p
8530 FOR p=1 TO 21
8535 IF p=4 THEN LET p=10
8540 IF p=13 THEN LET p=19
8545 LET q=a(p)+a(p+3)+a(p+6)
8550 GO SUB a
8555 NEXT p
8560 FOR p=1 TO 25 STEP 3
8565 LET q=a(p)+a(p+1)+a(p+2)
8570 GO SUB a
8575 NEXT p
8580 FOR p=1 TO 3
8585 LET q=a(p)+a(p+12)+a(p+24)
8590 GO SUB a
8595 NEXT p
8600 FOR p=1 TO 7 STEP 3
8605 LET q=a(p)+a(p+10)+a(p+20)
8610 GO SUB a
8615 NEXT p
8620 FOR p=7 TO 9
8625 LET q=a(p)+a(p+6)+a(p+12)
8630 GO SUB a
8635 NEXT p
8640 FOR p=3 TO 9 STEP 3
8645 LET q=a(p)+a(p+8)+a(p+16)
8650 GO SUB a
8655 NEXT p
8660 FOR p=1 TO 19 STEP 9
8665 LET q=a(p)+a(p+4)+a(p+8)
8670 GO SUB a
8675 NEXT p
8680 FOR p=3 TO 21 STEP 9
8685 LET q=a(p)+a(p+2)+a(p+4)
8690 GO SUB a
8695 NEXT p
8710 LET q=a(1)+a(14)+a(27)
8715 GO SUB a
8720 LET q=a(3)+a(14)+a(25)
8725 GO SUB a
8730 LET q=a(7)+a(14)+a(21)
8735 GO SUB a
8740 LET q=a(9)+a(14)+a(19)
8745 GO SUB a
8750 REM IS GAME OVER?
8760 LET go=go+1
8770 IF go=27 THEN GO TO 9000
8800 REM GO TO NEXT MOVE
8810 PRINT AT 22,0,,
8820 LET m=ABS (m-1)
8830 GO TO 1000-(800 AND m)
9000 REM COUNT COMPLETED LINES
9010 LET a=0: LET b=0
9020 FOR f=1 TO 49
9030 IF b(f)=3 THEN LET a=a+1
9040 IF b(f)=12 THEN LET b=b+1
9050 NEXT f
9200 REM PRINT FINAL SCORE
9210 PRINT AT 1,25;"You";AT 3,25
;"have";AT 5,25;b;AT 7,25;"line"
+("s" AND b<>1);"."
9220 PRINT AT 11,25;"I have";AT
13,25;a;AT 15,25;"line"+("s" AND
 a<>1);"."
9230 IF a>b THEN GO TO 9300
9240 IF b=a THEN GO TO 9400
9250 PRINT AT 22,0;"You win, cur
se it. Another game?"
9260 BEEP 1,-12
9270 GO TO 9430
9300 PRINT AT 22,0; FLASH 1;"I W
IN AGAIN!";
9310 PRINT " Another try,sucker?
"
9320 FOR g=1 TO 5: FOR f=12 TO 3
5 STEP 10: BEEP .05,f: NEXT f: N
EXT g
9330 GO TO 9430
9400 PRINT AT 22,0;"      "; INVE
RSE 1;"DRAW";
9410 PRINT "    Another game?"
9420 BEEP .5,24
9430 GO SUB 9500
9440 IF a$<>"Y" AND a$<>"N" THEN
 BEEP .1,-24: GO TO 9430
9450 PRINT AT 22,0,,
9460 POKE 23659,2
9470 IF a$="Y" THEN RUN
9480 FOR f=12 TO 0 STEP -1: BEEP
 .05,f: NEXT f: STOP
9500 REM INKEY$ ROUTINE
9510 IF INKEY$<>"" THEN GO TO 95
10
9520 IF INKEY$="" THEN GO TO 952
0
9530 LET a$=INKEY$
9540 RETURN
9600 REM STORE STATE OF LINES
9610 LET b(b)=q
9620 IF q=12 OR q=3 THEN BEEP .0
1,16+(12 AND q=3)
9630 LET b=b+1
9640 RETURN
9900 REM INSTRUCTIONS
9910 BORDER 4: PAPER 4: INK 0: F
LASH 0: BRIGHT 0: CLS
9920 PRINT AT 0,3; INVERSE 1;"3-
D NOUGHTS AND CROSSES"
9930 PRINT ''"The object of the
game is to    complete as many l
ines as you   can. The computer
will also try!"
9940 PRINT '"All straight lines
count,whetherhorizontal,vertical
 or diagonal.There are 49 possib
le lines."
9950 PRINT '''"Choose a skill le
vel:-"''TAB 4;"Press: 1 = dead e
asy",TAB 11;"2 = hard",TAB 11;"3
 = even harder"
9960 BEEP .05,40
9970 GO SUB 9500
9980 IF a$<"1" OR a$>"3" THEN BE
EP .1,-24: GO TO 9970
9990 LET level=VAL a$
9995 CLS : RETURN
```

H T Davis

AIR RAIDERS

Overhead excitement in this war-time simulation game with a difference.

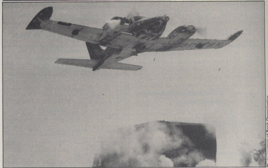


Photo courtesy of Rank Organisation.

This program differs a little bit from the normal 'war time' simulation games in that the shells actually follow a parabolic trajectory. The equation used to work out their motion is calculated in line 75 and simulates a projectile under gravity.

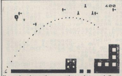
WHAT GOES UP…

The initial velocity and angle of projection can either be selected by the player (mode x) or by the computer (mode z). The aim of the game is to destroy the overhead aircraft, launched ICBMs and the ICBM base; the ICBMs and their base must be destroyed using shells and the aircraft should be destroyed using rockets. The game stops either after eight cycles or if you accidentally destroy an allied aircraft (you can spot them because they're blue!).

Full instructions are included in the program, including when one can fire a shell or a rocket. The program occupies virtually all of the 16K memory.

A VARIABLE SITUATION

The variables used in this program are:



Your rockets destroy the enemy plane — but your shells missed the enemy missiles.

g — The number of game cycles.
p — The game mode.
s — The score.
r — The rocket drive.
v — The rocket control.
w — The random element in the aircraft and drive.
x — The aircraft and ICBM drive.
(x,y) — The shell trajectory pixels.
(q,m) — The shell trajectory co-ordinates.
z — The shell control.
U (V on display) — The shell launch velocity.
A — The shell launch angle.

The variables used in the program.

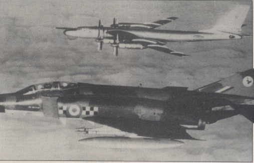


Photo courtesy of Associated Press Ltd.

```
   1 GO SUB 1000
   5 PAPER 6: INK 1
  10 DIM i$(704): PRINT AT 0,0;i
$
  12 BORDER 4
  14 PRINT INK 2; FLASH 1;AT 1,1
1;" AIR-RAID ": PRINT AT 4,2;"De
stroy the ICBM's and their       b
ase with shells (Key S)";AT 8,2;
"Destroy enemy aircraft with
 shells (S) and rockets (R)";AT
12,2;"Allied aircraft are blue a
nd     must not be hit"
  16 PAUSE 200: BORDER 5
  18 PRINT INK 2;AT 16,6;"To se
lect own shell           traject
ory, press (X)";AT 20,2;"For ran
dom launch, press (Z)"
  19 PAUSE 200
  20 LET s=0
  25 LET g=0
  30 IF INKEY$="x" THEN LET p=1
 GO TO 600
  32 IF INKEY$="z" THEN LET p=0
 GO TO 725
  34 GO TO 12
  40 LET v=0
  45 LET z=0
  50 LET w=INT (RND*7)
  60 FOR x=0 TO 60
  61 IF x<40 AND z=0 AND v=0 THE
N PRINT INK 0; PAPER 7; FLASH 1;
AT 0,1;"FIRE 5";AT 0,8;"V=";AT 0
,13;" ";AT 0,14;"A=";AT 0,10;U;"
";AT 0,16;A;""
  70 IF z=0 THEN LET y=0
  75 IF z>.5 AND x-z<32 THEN LET
 y=((x-z)/3)*TAN (A*PI/180)-(55*
(x-z)*(x-z))/(U*U*COS (A*PI/180)
*COS (A*PI/180))
  76 IF x-z>1 AND y<=.025 THEN G
O SUB 500
  78 IF y<=4 AND x-z<32 AND z>=1
 THEN PLOT 8*(x-z),40*y
  80 LET r=22-2*x+2*v
  85 IF v<>0 AND x-v<12 THEN PRI
NT INK 0;AT r,5;"B";AT r,3;"B":
REM GRAPHICS B
  86 IF v<>0 AND x-v>1 AND x-v<1
3 THEN PRINT AT r+2,5;" ";AT r+2
,3;" "
  90 IF x<31 THEN PRINT INK 1;AT
 2,31-x;"A": REM GRAPHICS A
  91 IF x>0 AND x<32 THEN PRINT
AT 2,32-x;" "
 100 IF 2*w+x<52 AND 2*w+x>20 TH
EN PRINT ; INK 4;AT 1,51-2*w-x;"
A"
 101 IF 2*w+x<53 AND 2*w+x>21 TH
EN PRINT AT 1,52-2*w-x;" "
 102 IF x-5*w>0 AND x-5*w<33 THE
N PRINT INK 1;AT 2,32+5*w-x;"A"
 103 IF x-5*w>1 AND x-5*w<34 THE
N PRINT AT 2,33+5*w-x;" "
 105 IF x-5*w>=0 AND x-5*w<31 TH
EN PRINT INK 2;AT 5,x-5*w+1;"E":
 REM GRAPHICS E
 106 IF x-5*w>0 AND x-5*w<32 THE
N PRINT AT 5,x-5*w;" "
 108 IF w+x>30 AND w+x<62 THEN P
RINT INK 2;AT 1,61-(w+x);"A"
 109 IF w+x>31 AND w+x<=62 THEN
PRINT AT 1,62-(w+x);" "
 110 IF x>24 AND x<56 THEN PRINT
 INK 3;AT 3,x-24;"E"
 111 IF x>25 AND x<57 THEN PRINT
 AT 3,x-25;" "
 113 IF x>29 THEN PRINT INK 4;AT
 5,x-29;"E"
 114 IF x>30 THEN PRINT AT 5,x-3
0;" "
 120 IF x>8 AND x<=29 THEN PRINT
 INK 3;AT 29-x,24;"C";AT 29-x,20
;"C": REM GRAPHICS C
 121 IF x>8 AND x<=30 THEN PRINT
 AT 30-x,24;" ";AT 30-x,20;" "
 124 IF 2*w+x>14 AND 2*w+x<=35 T
HEN PRINT INK 3;AT 35-(2*w+x),22
;"C"
 125 IF 2*w+x>14 AND 2*w+x<=36 T
HEN PRINT AT 36-(2*w+x),22;" "
 128 IF x>38 AND x<=59 THEN PRIN
T INK 5;AT 59-x,24;"C";AT 59-x,2
0;"C"
 129 IF x>38 AND x<61 THEN PRINT
 AT 60-x,24;" ";AT 60-x,20;" "
 132 IF w+x>37 AND w+x<59 THEN P
RINT INK 4;AT 58-(w+x),22;"C"
 133 IF w+x>37 AND w+x<60 THEN P
RINT AT 59-(w+x),22;" "
 150 LET q=INT ((175-40*y)/8): L
ET m=x-z
 155 IF ATTR (q,m)=57 AND m<32 A
ND z>.5 AND q>0 THEN PRINT AT q,
m;"G": GO TO 850: REM GRAPHICS G
 156 IF ATTR (q,m)>57 AND m<32 A
ND z>.5 AND q>0 THEN PRINT AT q,
m;"G": BEEP 1,-25: LET s=s+200
 160 IF SCREEN$ (q,m)="'" AND m<
32 AND z>.5 THEN PRINT AT q,m;"
": BEEP .2,-10: LET s=s+50
 170 IF ATTR (r,3)=57 AND v<>0 A
ND x-v<12 THEN PRINT INK 6;AT r,
3;"G": BEEP 1,-20: GO TO 850
 171 IF ATTR (r,5)=57 AND v<>0 A
ND x-v<12 THEN PRINT INK 6;AT r,
5;"G": BEEP 1,-20: GO TO 850
 173 IF ATTR (r-1,3)>57 AND v<>0
 AND x-v<11 THEN PRINT INK 6;AT
r-1,3;"G": BEEP 1,-20: LET s=s+1
20
 174 IF ATTR (r-1,5)>57 AND v<>0
 AND x-v<11 THEN PRINT INK 6;AT
r-1,5;"G": BEEP 1,-20: LET s=s+1
00
 180 PRINT INK 0; FLASH 1;AT 0,2
8;s;""
 200 IF x<40 AND INKEY$="s" THEN
 GO SUB 300
 205 IF INKEY$="r" THEN GO SUB 4
20
 215 IF x>=40 AND v=0 THEN GO SU
B 303
 220 IF v=0 AND z>.5 THEN GO SUB
 305
 230 IF x-v>11 AND z>0 THEN GO S
UB 305
 235 IF x-v>11 AND z=0 THEN GO S
UB 550
 240 NEXT x
 245 DIM i$(25): PRINT AT 0,0;i$
 250 DIM i$(445): PRINT AT 1,0;i
$
 255 DIM i$(29): PRINT AT 14,0;i
$: PRINT AT 15,0;i$: PRINT AT 16
,0;i$: PRINT AT 17,0;i$
 260 DIM i$(17): PRINT AT 18,0;i
$: PRINT AT 19,0;i$
 261 DIM i$(16): PRINT AT 20,1;i
$
 263 LET g=g+1: IF g=8 THEN GO T
O 900
 265 IF p=1 THEN GO TO 605
 268 IF p=0 THEN GO TO 725
 270 GO TO 40
 300 LET z=x
 303 IF INKEY$="r" THEN GO TO 40
2
 306 IF v=0 OR x-v>9 THEN PRINT
INK 0; FLASH 1; PAPER 6;AT 0,0;"
 PRESS R ": PRINT PAPER 7;AT 0,9
;"       "
 310 RETURN
 400 IF z<.5 THEN GO TO 450
 402 IF v=0 AND x-z>4 THEN LET v
=x
 405 IF x-v>11 THEN LET v=x
 445 IF x=v THEN PRINT PAPER 7;A
T 0,0;"                         "
 450 RETURN
 500 IF z<>0 THEN LET z=.5
 505 RETURN
 550 IF v<>0 THEN GO TO 305
 555 RETURN
 600 CLS
 603 IF g=0 THEN GO TO 780
 605 INPUT "Enter Speed V",U
 610 INPUT "Enter Angle A",A
 615 GO TO 40
 725 LET U=60+INT (RND*40)
 730 LET A=25+INT (RND*40)
 740 IF g<>0 THEN GO TO 40
 780 CLS
 790 BORDER 5: INK 0: PAPER 7
```

```
 795 DIM i$(704): PRINT AT 0,0;i
$
 800 PRINT INK 2;
    AT 14,29;"        █ ";
    AT 15,29;"        ▐▌";
    AT 16,29;"        ▐▌";
    AT 17,29;"        ▐▌";
    AT 18,17;"▐▌  ████";
    AT 19,17;"▐▌  ████";
    AT 20,17;"    ████"
 805 PRINT OVER 1; INK 2;
   AT 14,29;"  . . ";
   AT 15,29;"  . ..";
   AT 16,29;"  . . ";
   AT 17,29;"  . ..";
   AT 18,17;"....   . . . . .";
   AT 19,17;"....   .  .  . .";
   AT 20,17;"  .    . . . . ."
 810 PRINT INK 4;AT 21,2;"██ █
█████████████ █ █▄███████": PRINT
 INK 0;AT 21,3;"█";AT 21,5;"█"
;
 815 PRINT OVER 1; INK 4;AT 21,2
1;"'";AT 21,23;"'"
 830 PRINT AT 21,0;"H": REM GRAP
HICS H
 840 IF p=1 THEN GO TO 605
 845 IF p=0 THEN GO TO 40
 850 CLS
 854 INK 1
 860 PRINT FLASH 1;AT 5,11;"GAME
OVER": PRINT AT 8,4;"You have d
estroyed one";AT 10,6;"of your o
wn planes";AT 13,6;"Your score i
s "; FLASH 1; INK 0; PAPER 7;s;"
"
 865 PRINT AT 16,4;"Press <P> to
play again"
 870 FOR d=0 TO 7
 874 IF INT (d/2)=d/2 THEN BEEP
.5,2
 876 IF INT (d/2)<>d/2 THEN BEEP
.5,-2
 880 IF INKEY$="p" THEN GO TO 5
 885 NEXT d
 890 GO TO 860
 900 CLS
 910 INK 1
 920 PRINT FLASH 1;AT 3,11;"GAME
OVER": PRINT AT 8,6;"You have s
cored "; FLASH 1; INK 1; PAPER 7
;s;""
 925 PRINT AT 11,4;"without loss
of aircraft";AT 16,5;"Press <P>
to play again"
 930 FOR e=10 TO 70 STEP 10
 935 BORDER e/10
 940 IF SIN (2*e/PI)<.4 THEN BEE
P .5,10
 945 IF SIN (2*e/PI)>=.4 THEN BE
EP .6,-20
 950 IF INKEY$="p" THEN GO TO 5
 955 NEXT e
 960 GO TO 920
1000 DATA 0,96,96,98,255,98,96,9
6
1010 DATA 8,62,8,8,28,0,0,0
1020 DATA 24,24,24,24,24,60,60,3
6
1030 DATA 0,6,6,70,255,70,6,6
1040 DATA 126,189,219,231,231,21
9,189,126
1050 DATA 2,4,8,16,32,40,30,255
1100 LET u=PEEK 23675+256*PEEK 2
3676
1110 FOR i=0 TO 7: READ j: POKE
u+i,j: NEXT i
1120 FOR i=0 TO 7: READ j: POKE
u+8+i,j: NEXT i
1130 FOR i=0 TO 7: READ j: POKE
u+16+i,j: NEXT i
1140 FOR i=0 TO 7: READ j: POKE
u+32+i,j: NEXT i
1150 FOR i=0 TO 7: READ j: POKE
u+48+i,j: NEXT i
1160 FOR i=0 TO 7: READ j: POKE
u+56+i,j: NEXT i
1200 RETURN
```

ROSE SOFTWARE

EDUCATIONAL SOFTWARE FOR THE SPECTRUM (16K or 48K)

YOUNG LEARNERS 1

4 programs for the young computer user: abacus, telling the time, shopping and snakes. All with excellant graphics. An easy introduction to computer aided learning.

£5.95 per cassette

G.C.E. "O" LEVEL MATHS REVISION GEOMETRY

6 programs of multiple choice questions, covering properties of parallel lines, triangles, various polygons and circles. All questions use generated numbers. An explanation of the correct answer is given.

G.C.E. "O" LEVEL FRENCH VOCABULARY REVISION

Six programs on one cassette. The first 3 programs test vocabulary under various headings: eg "on holiday". Three further programs: homework tester, common words and phrases, and irregular verbs.

We have a large range of educational programs for the Spectrum and ZX81.

Send sae for catalogue of all our products to:
ROSE SOFTWARE,
148 Widney Lane, Solihull, West Midlands B91 3LH

SPECTRUM

DUAL FUNCTION JOYSTICK INTERFACE £13·95INC

WHY PAY £35 FOR 2 INTERFACES WHEN YOU CAN BUY OUR DUAL INTERFACE FOR JUST £13.95 INC.

FEATURES

- ★Compatible with any software that list the Kempston Joystick.
- ★Compatible with any software that uses cursor keys.
- ★Compatible with most Psion/Melbourne House games e.g. Terror Daktil, Horace Goes Ski-ing etc.
- ★Allows the use of Atari/Le Stick type joystick.
- ★Eight directions can be programmed in simple basic.
- ★Free draw programme and full instructions supplied.

STANDARD INTERFACE FEATURES

Same as Dual Interface

- ★But only compatible with any software that lists the Kempston Joystick

ZX-81/SPECTRUM

LOADING PROBLEMS SOLVED WITH

The **CODER** as reviewed in the Sinclair User August '83 **£9.95** INC

DUAL INTERFACE .... £13.95 inc
STANDARD INTERFACE .... £10.95 inc

£1 discount if joystick ordered with any interface e.g. Dual/Int & Joystick £19.45 inc

Send Cheques/P.O. to:
JILES ELECTRONICS
48 Parkway, Chellaston, Derby DE7 1QZ. Tel: (0332 703892)
Overseas orders add £2.00

48K SPECTRUM

1984
A GAME OF GOVERNMENT MANAGEMENT
48K SPECTRUM CHALLENGE FROM INCENTIVE SOFTWARE LTD

1984

THE GAME OF ECONOMIC SURVIVAL

THE BRITISH ECONOMY WITH YOU AT THE CONTROLS! WHAT SORT OF CHANCELLOR WOULD YOU MAKE WITH SEVERAL BILLION POUNDS TO SPEND & FIVE YEARS TO THE NEXT GENERAL ELECTION? GRAPHIC DISPLAYS, HISTOGRAMS & AN ANNUAL PERFORMANCE RATING ARE ALL INCLUDED TO SHOW HOW YOU ARE DOING. HOW MANY YEARS WILL YOU LAST?

FREE INSIDE: Pocket Guide to Running Britain !

N003 £5.50

MOUNTAINS OF KET

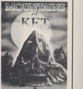
ADVENTURE

A MONSTER OF AN ADVENTURE PROGRAM! COMBAT, INTER–ACTIVE BEINGS, MONETARY SYSTEM, MAGIC, EDGAR, SAVE/LOAD FACILITY PLUS MANY OTHER FEATURES.

As well as being a fast ingenious compelling adventure in itself the Mountains of Ket is the first of a 3 part series that builds into a mammoth adventurers challenge.

Incentive: It could be adventageous if you achieve 100%!!

N002 £5.50



SPLAT!
48K SPECTRUM CHALLENGE FROM INCENTIVE SOFTWARE LTD

SPLAT!

ARCADE

ONE OF THE MOST ORIGINAL AND COMPELLING ARCADE GAMES EVER PRODUCED! STARRING ZIPPY!!

"SPLAT! is one of the most addictive games I have ever played on the 48K SPECTRUM. It is certainly the most original"
Computer & Video Games

NOW AVAILABLE FROM WH SMITH AND BOOTS.

N001 £5.50

All programs run in the **48K ZX SPECTRUM** and are available from all good computer shops. In case of difficulty please order direct using the coupon below.

Please send me (tick box(es) required)

1984 ☐ MOUNTAINS OF KET ☐ SPLAT ☐

All at £5.50 each (inclusive of VAT and 1st class postage)

I enclose cheque/P.O. for £ or debit my Access Account No.

Name

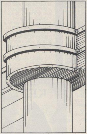
Address

INCENTIVE SOFTWARE LTD., 54 London Street, Reading RG1 4SQ. Tel: Reading (0734) 591678

Colin Gooch

DRAINPIPES

Here is an adaption of an old arcade game for you to try your hand at.



Drainpipes is based on the old style of mechanical arcade game in which steel balls were dropped into cups or tubes.

The basis of the game is simple enough. A ball runs along a stepped roofline at the top of the screen and will drop down if you press 'D'. Below it are a series of seven 'drainpipes' and you score by making it drop into the centre of one of these. Each drainpipe will only score once and a red indicator will light up to show where a score has been made. A score in each pipe will give a maximum of 28 points.

This version of the game has a random 'flip' feature. A red and white flag will sometimes illuminate as the ball is dropping and at the same time, the rate of drop slows down. If you don't like where the ball is heading press 'K' and it will slide off to the left and make its way back to the start without incrementing the ball score. You can then try again. Alternatively, as it slides off to the left you can press 'D' again whereupon it will once more go on its downward path. After 12 balls (or a maximum score!) the computer will tell you your score and invite you to play again. If you don't want to continue, then pressing 'N' will get you a fond farewell.

DOWN THE DRAIN

The main movement of the ball and the setting of the graphics is done in lines 20 to 320. Lines 350 to 400 detect a scoring ball and label it with a score value. Lines 460 to 480 check to see if there is a full score line, ie that no tube still has a score of zero, and if necessary terminates the round by allowing the next loop 500 to 530 to add up the score. These lines also add the score if you run out of balls.

Lines 600 to 660 print out the score and start a new game. LInes 670 to 690 terminate the game. The 'flip' routine is in lines 710 to 890 and shift the ball around the screen as required. Lines 900 onward print out the game instructions.

VARIATIONS ON A THEME

Anyone who wishes can find a lot of variations that can be done with this program. For a simple game, you can omit the 'flip' routine by leaving out line 190 and lines 710 to 890. You can also omit line 15 and the instructions.,

Line 100 looks a little clumsy, but it works! It could be done with a subroutine of print statements. It can also be made straight . . . though whether this makes for an easier or harder game is a matter of opinion.

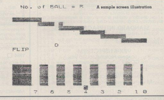
You can make things a little more difficult by reducing 'ra' in line 190 thus getting less flips. You could also reduce the PAUSEs in lines 730 and 800 to increase the speed of each ball.

And if you really want to get your fingers confused on the keyboard, why not write an extra subroutine called by INKEY$ "L" which will cause the ball to slide to the right. If you do this, the subroutine would be called from line 740.

A GRAPHIC EXPLANATION

To help you type the program in, Colin has provided us with an explanation of some of the lines.

Line 100 includes seven Graphics shifted 8s followed by 31 spaces, then six Graphics shifted 8s followed by 31 spaces.
Line 240 includes four Graphics shifted 8s, one Graphics shifted 5, one Graphics 8, one Graphics 5, one Graphics shifted 8, one Graphics shifted 5, and so on to until the end of the line which you should finish with a Graphics 5.
Line 360 includes one Graphics shifted 8.
Line 840 includes three Graphics shifted 8s.



A sample screen illustration

```
   5 REM DRAINPIPES
  10 REM BY C.N.GOOCH ©1982
  15 CLS : GO SUB 900
  20 LET ps=0: LET pb=12
  30 BORDER 5: PAPER 6: CLS : LE
T sc=0
  40 LET ba=0: LET bb=0
  50 DIM c(7)
  60 LET p=2
  80 PRINT AT 20,0; PAPER 4,"
 7   6   5   4   3   2   1  0"
 100 LET a$="████████
██████████
████████"
 110 PRINT AT 3,0; INK 2;a$
 120 GO SUB 240: REM SET UP ROOF
 130 LET a=0: LET b=2: PAUSE p*5
0
 140 PRINT AT b,a; INK 3;"O": P
RINT AT b-1,a;" ": IF a=0 THEN B
EEP .15,-20: REM START BALL
 150 LET ra=1+INT (RND*10): LET
a=a+1: IF a=7 OR a=12 OR a=17 OR
 a=22 OR a=27 THEN LET b=b+1
 160 IF a=31 THEN GO TO 190
 170 IF a>3 AND INKEY$="d" OR A>
3 AND INKEY$="D" THEN BEEP .15,1
0: GO TO 190
 180 GO TO 140
 190 IF ra<4 THEN GO TO 710: REM
 DROP BALL
 200 PRINT AT b,a; INK 3;"O": PR
INT AT b-1,a;" ": LET b=b+1
 210 IF b=19 THEN LET b=19: GO S
UB 350: GO TO 230
 220 GO TO 190
 230 LET p=1+INT (RND*3): GO SUB
 250: GO TO 130
 240 LET b$="██  █   █   █   █
█   █   █   █"
 250 FOR n=1 TO 5: REM SET UP PI
PES
 260 PRINT AT 13+n,0; INK 1;b$
 270 NEXT n
 290 PRINT AT 3,0; INK 2;a$
 300 LET ba=ba+1: PRINT AT 0,2;"
No. of BALL = ";ba: IF ba=13 THE
N GO SUB 500: PAUSE 100: GO TO 6
00
 310 GO SUB 460
 320 IF sc>=28 THEN PRINT AT 0,1
8;"Score= ";sc;AT 1,8; INK 2; PA
PER 7; FLASH 1;"M A X I M U M ":
 PAUSE 200: GO TO 600
 330 RETURN
 350 FOR i=5 TO 29 STEP 4: REM I
NDICATE SCORE
 360 IF b=19 AND a=i THEN PRINT
AT 19,i; INK 2; FLASH 1;"█"
 370 IF a=i AND c((33-i)/4)=0 TH
EN BEEP .15,50
 380 IF a=i THEN LET c((33-i)/4)
=(33-i)/4
 390 NEXT i
 400 RETURN
 460 FOR q=1 TO 7: REM DETECT FU
LL SCORELINE
 470 IF c(q)=0 THEN RETURN
 480 NEXT q
 500 FOR l=1 TO 7: REM ADD SCORE
S
 510 LET sc=sc+c(l)
 520 NEXT l
 530 RETURN
 600 CLS : PRINT AT 3,5; INK 1;
PAPER 6;" YOUR SCORE WAS ";sc;AT
 5,7;" IN ";(ba-1);" BALLS "
 610 IF (ba-1)<pb THEN LET pb=(b
a-1)
 620 IF sc>ps THEN LET ps=sc
 625 PRINT AT 6,4; PAPER 1; INK
7;" HIGHEST SCORE SO FAR ";ps;AT
 7,10;"IN ";pb;" BALLS "
 630 PRINT PAPER 4;AT 12,0;"   D
O YOU WANT ANOTHER GO?       "("
       PRESS"; PAPER 7; INK 2;"Y"
; PAPER 4; INK 0;"ES OR "; PAPER
 7; INK 2;"N"; PAPER 4; INK 0;"O
"
 640 IF INKEY$="Y" OR INKEY$="y"
 THEN GO TO 25
 650 IF INKEY$="n" OR INKEY$="N"
 THEN GO TO 670
 660 GO TO 640
 670 CLS
 675 PRINT TAB bb+5; PAPER 1+INT
 (RND*7); INK 9; BRIGHT 2-(1+INT
 (RND*2));" THANKS FOR PLAYING,
BYEE E  E   E   E ": IF bb>55 THE
N GO TO 690
 680 LET bb=bb+1: POKE 23692,255
: BEEP .25,bb: GO TO 675
 690 CLS : PRINT AT 4,4; PAPER 6
; INK 2; BRIGHT 1; FLASH 1;"  A
N Y O N E  W A N T  A  G A M E
?  "'TAB 4;"  PRESS ENTER TO ST
ART  ";AT 15,7; INVERSE 1;"D R A
 I N P I P E ": INPUT Q$: IF Q$=
"" THEN GO TO 25
 710 IF b<13 THEN PRINT AT 10,0;
 INK 2; PAPER 7; FLASH 1; BRIGHT
 1;"FLIP"
 720 IF b=13 THEN PRINT AT 10,0;
 PAPER 6; FLASH 0; BRIGHT 0;"
"
 730 PRINT AT b,a; INK 3;"O";AT
b-1,a;" ": PAUSE 5
 740 IF b>10 AND b<14 AND  INKEY
$="k" OR INKEY$="K" THEN GO TO 8
00
 760 LET b=b+1
 770 IF b>=14 THEN GO TO 210
 780 GO TO 710
 800 PRINT AT b,a; INK 3;"O ": P
AUSE 2
 810 LET a=a-1: IF a=2 THEN PRIN
T AT 10,0;"    ": IF a=2 THEN GO
 TO 830
 815 IF INKEY$="d" OR INKEY$="D"
 THEN GO TO 870
 820 GO TO 800
 830 PRINT AT b,a; INK 3;"O ";AT
 b+1,a;" ": IF b=12 THEN PRINT A
T 14,0; INK 1;b$
 840 LET b=b-1: IF b=2 THEN PRIN
T AT 10,0;"    ";AT 3,2; INK 2;"
██"
 850 IF b=2 THEN GO TO 140
 860 GO TO 830
 870 PRINT AT b,a; INK 3;"O ";AT
b-1,a;" ";AT 10,0;"    "
 880 LET b=b+1: IF b=19 THEN LET
b=19: GO SUB 350: GO TO 230
 890 GO TO 870
 900 BORDER 6: PRINT PAPER 5; IN
K 2;AT 1,5;" D R A I N P I P E S
"
 920 PRINT PAPER 7;AT 3,0;"  YOU
 WILL SEE A BALL ROLLLING   ALON
G A ROOFTOP.  YOU MUST GET IT TO
 DROP INTO A DRAINPIPE BY  PRESS
ING ""D""                       "
 930 PRINT AT 6,0; PAPER 7;" YOU
 CAN ONLY REGISTER A SCORE  ONCE
 IN A PIPE, A RED INDICATOR SHOW
S WHERE THERE HAS BEEN A    SCO
RE                            "
 940 PRINT AT 13,0; INVERSE 1;"
IF A FLAG "; INK 2;" FLIP "; PAP
ER 7; INK 0;"SHOWS ON THE   LEFT
  PRESSING ""K"" WILL CAUSE
THE BALL TO MOVE TO THE LEFT   "
 950 PRINT AT 17,0; PAPER 7;" TH
E BALL WILL RETURN TO THE    STA
RT UNLESS YOU PRESS ""D"" AGAIN.
  IF YOU DO THE BALL WILL DROP I
N THE NEW PLACE                "
 960 INPUT "PRESS ENTER TO START
";Z$: IF Z$="" THEN GO TO 20
```

Robert Erskine

MACHINE CODE COLOUR GRAPHICS

We show you how to speed up your colour graphics on the ZX Spectrum.

ZX Spectrum

Although Spectrum BASIC is fairly fast, there are many occasions when it is not nearly fast enough; particularly when you need to print large or multiple images to the screen.

The most efficient way of solving this problem is to use machine code routines to do the job, for the speed of the Spectrum's microprocessor is such that several graphics, or even the entire screen, can be printed in a small fraction of a second. Although there is a routine in the ROM which enables you to do this (using an RST 10 instruction) the process can be rather long winded unless each character in the graphic follows the previous one on the screen. Furthermore, since this routine draws its graphics symbols from existing memory, you will always be confined to using the standard character set and user defined graphics.

What would be ideal would be a program which enables you to 'read' every byte of your most complex graphic masterpieces and store them away in RAM for instant recall to any position on the screen. The two programs in this article, Grafcode and Grafprint, are designed to do exactly that.

DYNAMIC DUO

Grafprint is a machine code program which builds up a graphic image on the screen, including attributes, from a data file held in memory. It can either be called from within a BASIC program or incorporated in a larger machine code program where it can be used to greater effect in generating laser fire or swooping aliens.

The Demo program illustrates Grafprint at work, printing a large green moon lander at the top left of the screen. By altering lines 20 and 30 which hold the low and high byte values of the first attribute position on the screen, you will see that the image can be printed anywhere, in spite of the peculiar design of the Sinclair display file.

Line 90 of Demo holds the machine code data of Grafprint and line 100 holds the data for the graphic.

The Grafprint program is designed to be loaded from address 32300 to address 32411 and therefore RAMtop should be at 32299 or less. The first eight bytes are used as temporary pigeon holes by the main program, which starts at

32308. Although Grafprint can be moved elsewhere in memory, it contains references to these first eight bytes and these would therefore have to be changed.

Grafcode is a BASIC program which generates data files of graphics for Grafprint. To use it, first create your graphic image (or text) anywhere on the screen, preferably from the top left-hand corner where you can find the start address easily. The image may be created in the normal way, using BASIC colour commands, user defined graphics and so on. Then, when it is ready, activate Grafcode by entering the command GO TO 9600. On no account RUN the program or your splendid new handiwork will disappear without trace.

Grafprint disassembled

```
Addr   Hex Code  Mneumonic
7E2C   00
7E2D   58
7E2E   9C
7E2F   7E
7E30   000000
7E33   00
7E34   2A2E7E    LD     HL, (7E2Eh)
7E37   EB        EX     DE,HL
7E38   2A2C7E    LD     HL,(7E2Ch)
7E3B   1A        LD     A,(DE)
7E3C   32307E    LD     (7E30h),A
7E3F   13        INC    DE
7E40   1A        LD     A,(DE)
7E41   77        LD     (HL),A
7E42   D5        PUSH   DE
7E43   22327E    LD     (7E32h),HL
7E46   EB        EX     DE,HL
7E47   21FF58    LD     HL,58FFh
7E4A   A7        AND    A,A
7E4B   ED52      SBC    HL,DE
7E4D   380B      JR     C, + 0Bh
7E4F   2A327E    LD     HL,(7E32h)
7E52   110018    LD     DE,1800h
7E55   A7        AND    A,A
7E56   ED52      SBC    HL,DE
7E58   1820      JR     + 20h
7E5A   ED5B327E  LD     DE,(7E32h)
7E5E   21FF59    LD     HL,59FFh
7E61   A7        AND    A,A
7E62   ED52      SBC    HL,DE
7E64   380B      JR     C, + 0Bh
7E66   2A327E    LD     HL,(7E32h)
7E69   110011    LD     DE,1100h
7E6C   A7        AND    A,A
7E6D   ED52      SBC    HL,DE
7E6F   1809      JR     + 09h
7E71   2A327E    LD     HL,(7E32h)
7E74   11000A    LD     DE,0A00h
7E77   A7        AND    A,A
7E78   ED52      SBC    HL,DE
7E7A   D1        POP    DE
7E7B   13        INC    DE
7E7C   0608      LD     B,08h
7E7E   1A        LD     A,(DE)
7E7F   77        LD     (HL),A
7E80   05        DEC    B
7E81   2804      JR     Z, + 04h
7E83   24        INC    H
7E84   13        INC    DE
7E85   18F7      JR     - 09h
7E87   2A327E    LD     HL,(7E32h)
7E8A   3A307E    LD     A,(7E30h)
7E8D   0601      LD     B,01h
7E8F   90        SUB    A,B
7E90   C8        RET    Z
7E91   32307E    LD     (7E30h),A
7E94   13        INC    DE
7E95   1A        LD     A,(DE)
7E96   0600      LD     B,00h
7E98   4F        LD     C,A
7E99   09        ADD    HL,BC
7E9A   18A3      JR     - 5Dh
7E9C   0138F0    LD     BC,F038h

Control Bytes
```

WHAT'S IN STORE?

You will then be prompted to input the address from which you want the graphics data to be stored, the first attribute address currently occupied by the graphic and the number of character squares in the graphic. You will then be asked to input each of the displacements between the characters. These refer to attribute file displacements so that one square to the right = 1, one square below = 32, and so on. You may then sit back and hum a little tune for a few seconds whilst Grafcode translates your handiwork into a sequential data file which can be accessed by Grafprint.

Having loaded Grafprint and prepared you data file, the next step is to load the start address of the data file into 32302/3 and the attribute address of the chosen screen location into 32300/1. Obviously in a moving graphics program, this would be done in machine code. To print the graphic to the screen, simply enter the command RANDOMIZE USR 32308.

Grafprint works by moving a pointer around the attributes file of the Spectrum and loading the attributes of the current character into an address, followed by the corresponding eight display file bytes for each character. The data file which it reads is formatted so that each character is represented by 10 bytes. The first byte holds the displacement between the current character and the one before, the second holds the attributes code and the remaining eight hold the graphic image. The data for the first character in a graphic also uses 10 bytes but instead of staring with a displacement value, it holds the total number of characters in the graphic.

The program incorporates a routine which calculates which of the three screen zones the current character will occupy and selects the corresponding display addresses accordingly. This ensures that if the graphic crosses a border between zones ▶

then the relative positions of the characters are maintained.

THE SKY'S THE LIMIT

There is no limit to the size of the data file which can be used by Grafprint, other than the size of your machine's memory, and if files are built up section by section, there is no limit to the different shapes you can store. Having used 21 user defined graphics in a picture you can convert it to data using Grafcode, store it temporarily on tape and continue with a new set of graphics, bringing the whole lot together in one file when you have finished. Graphics involving circles and other shapes can equally be stored.

If you intend to move graphics rapidly around the screen using Grafprint, it will be necessary to erase each preceding image before printing the next. One way of doing this is to use a machine code equivalent of PRINT OVER, say by holding a blank graphic of the same size and shape as the original and using Grafprint to print it over the top. In some cases, it will be sufficient to clear the screen between each printing because the speed of machine code is such that the illusion of continuous action will be maintained. Screen clearing can be achieved by filling all the display file addresses with zeros and all the attribute addresses with an appropriate code. If the background to your moving graphics is complex and you don't wish to clear it between moves, you can store a complete copy of the screen above RAMtop using a block move routine in machine code and reprint it to the screen between moves by means of a similar routine. Since this technique uses up nearly seven thousand bytes of RAM it is hardly worth contemplating on a 16K Spectrum, unless the rest of your program is entirely in machine code.

The program listings

```
   1 REM GRAFCODE
9600 PRINT AT 21,0;"Enter data s
tart"
9605 INPUT data: LET dat=data
9610 PRINT AT 21,0;"Enter first
att address"
9615 INPUT att
9620 PRINT AT 21,0;"How many cha
racters in graphic?"
9625 INPUT char: DIM d(1): IF ch
ar>1 THEN DIM d(char-1)
9630 POKE data,char: LET data=da
ta+1
9632 IF char=1 THEN GO TO 9655
9635 FOR x=1 TO char-1
9637 PRINT AT 21,0;"
                   "
9640 PRINT AT 21,0;"Displacement
 ";x;"?
                   "
9645 INPUT d(x)
9650 NEXT x
9655 FOR x=1 TO char
9660 POKE data,PEEK att
9665 IF att<22784 THEN GO TO 969
5
9670 IF att<23040 THEN GO TO 968
5
9675 LET byte=att-2560
9680 GO TO 9700
9685 LET byte=att-4352
9690 GO TO 9700
9695 LET byte=att-6144
9700 LET count=0
9705 LET data=data+1
9710 POKE data,PEEK byte
9715 LET count=count+1
9720 IF count=8 THEN GO TO 9735
9725 LET byte=byte+256
9730 GO TO 9705
9735 LET data=data+1
9740 IF x<char THEN POKE data,d(
x)
9745 IF x<char THEN LET att=att+
d(x)
9750 LET data=data+1
9755 NEXT x
9760 PRINT AT 21,0;"Done-from ";
dat
9765 STOP
```

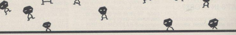
The listing of the Grafcode program.

```
Enter as a direct command:
RANDOMIZE USR 32308
```

The moon lander graphic printed when the Demo program is RUN.

```
   1 REM DEMO
  10 CLEAR 32299
  20 POKE 32300,0
  30 POKE 32301,88
  40 POKE 32302,156
  50 POKE 32303,126
  60 FOR x=32308 TO 32451
  70 READ byte: POKE x,byte
  80 NEXT x
  90 DATA 42,46,126,235,42,44,12
6,26,50,48,126,19,26,119,213,34,
50,126,235,33,255,88,167,237,82,
56,11,42,50,126,17,0,24,167,237,
82,24,32,237,91,50,126,33,255,89
,167,237,82,56,11,42,50,126,17,0
,17,167,237,82,24,9,42,50,126,17
,0,10,167,237,82,209,19,6,8,26,1
19,5,40,4,36,19,24,247,42,50,126
,58,48,126,6,1,144,200,50,48,126
,19,26,6,0,79,9,24,163
 100 DATA 4,60,63,127,225,255,25
5,127,63,31,1,60,252,254,135,255
,255,254,252,248,31,60,15,24,48,
95,255,128,128,128,1,60,240,24,1
2,6,255,1,1,1
 110 PRINT AT 20,0;"Enter as a d
irect command:";AT 21,0;"RANDOMI
ZE USR 32308"
```

The listing of the Demo program.



ONLY THE BEST BECOME A...

FIGHTER PILOT

A SPECTACULAR FLIGHT SIMULATION OF THE WORLD'S MOST EXCITING JET FIGHTER WITH STUNNING 3D COCKPIT VIEW AND THE MOST REALISTIC 3D AIR-TO-AIR COMBAT EVER SEEN ON THE SPECTRUM!

U.S. AIR FORCE

The SUPREME SIMULATION....

it leaves the rest at the

runway threshold!!

Based upon the F15 USAF air superiority jet fighter. Includes landing, take-off, steering on the ground, 3-D air to air combat, fully aerobatic performance, airfield attack by the enemy, crosswinds, turbulence, landing in fog, 4 pilot ratings, 4 runways, 8 beacons, ground features, Artificial Horizon, Radar, Flight Computer, ILS, flaps, u/c system failures, Map, joystick compatible — Kempston, AGF, Sinclair Interface 2, Protek & others. 100% machine code. Complete with full instructions and pilot's notes.

£7.95

48k Spectrum

Joystick compatible

DIGITAL Integration

Dept. , 22 Ash Church Road, Ash, Aldershot, Hants GU12 6LX

All prices include VAT and p&p (overseas add 55p per cassette) Trade & export enquiries welcome —

All cassettes carry a lifetime guarantee. Mail order or check your local computer store.

NIGHT GUNNER NEW

NIGHT GUNNER — the air battle scenario where only aces survive! Each mission pushes you to the limit with enemy fighters attacking from all directions, guns ablazing! — explosive 3D ground attack sortie, and there's **still** the flight home.

Joysticks — Kempston, AGF, Sinclair, Protek & others.

48k Spectrum £6.95

FOOTBALL League Analysis NEW

FOOTBALL League Analysis Program — The easy-to-use Pools Prediction Program with 9 methods of team performance analysis. Simply enter the new fixture list, load your league table data base, and let your Spectrum guide you to your Pools selection.

48k Spectrum £7.95

Taxi! NEW

TAXI! — Another passenger awaits at the busy town square. To the hotel? The station? The quicker you are, the bigger your tip and the less petrol you will use. But beware of reckless drivers — they simply hate taxis!!

A frustrating test of your skills as a taxi driver .... fun for all the family!!

Any Spectrum £5.50

Michael Guy

GETTING

Having seen the program, Lemonade stand, adapted for a variety of micros, Michael was a little disappointed not to find a version of the game for the ZX Spectrum. So, like any enterprising programmer, he sat down and wrote one for himself. The result of his work follows in the published listing — you should enjoy it, his computer club liked it so much they awarded him a small prize!
In the program, you are placed in charge of a lemonade stand in Hyde Park, London. You start off with a mere £10 and your object, in true capitalist fashion, is to make as much money as you can. Fortunately, there are only two major decisions you need make: how many cups to make that day and how much you intend to charge for each cup of lemonade.

Once this decision has been made, you will be greeted with a screen display of two glasses on the counter which slowly fill up. A clock appears on-screen to tell you the time of day, and you have to shut up the stand at seven o'clock.

At the end of each day, you are shown a report of the day's takings, and your profit and losses are added and subtracted from your total spending money. Should you run out of money, you will be told how well you did and if you manage to last out the whole week, you will be given some praise.

Take care of the vandals though, if they wreck your stand you'll have to fork out £20 to get the repairs done!

```
   1 REM
   3 BORDER 1: PAPER 1: INK 7
   4 PRINT AT 10,5;"Press 'ENTER
' to begin": INPUT z$
   5 CLS
   6 GO SUB 5000
   7 LET l=0: LET m=1000: LET r=
50: LET p=3
   8 LET day=1
   9 GO SUB 1000
  10 CLS : IF RND>.97 AND day>2
THEN GO TO 3000
  11 PRINT AT 1,2;"REPORT"
  20 PLOT 14,159: DRAW 52,0
  25 PRINT '"  DAY ";day
  30 DIM w$(6,16)
  31 LET w$(1)="cloudy"
  32 LET w$(2)="sunny"
  33 LET w$(3)="rainy"
  34 LET w$(4)="stormy"
  35 LET w$(5)="HOT"
  36 LET w$(6)="snowy"
  39 DIM a$(6,18)
  40 LET a$(1,17)=CHR$ 60
  41 LET a$(2,17)=CHR$ 80
  42 LET a$(3,17)=CHR$ 15
  43 LET a$(4,17)=CHR$ 7
  44 LET a$(5,17)=CHR$ 105
  45 LET a$(5,17)=CHR$ 2
  50 LET a$(1,18)=CHR$ 40
  51 LET a$(2,18)=CHR$ 90
  52 LET a$(3,18)=CHR$ 15
  53 LET a$(4,18)=CHR$ 10
  54 LET a$(5,18)=CHR$ 130
  55 LET a$(6,18)=CHR$ 8
 100 LET d=INT (RND*6+1)
 110 PRINT '" The weather today
is ";w$(d, TO 16)
 140 PRINT '" Lemonade Costs ";
;"p. per cup"
 150 PRINT '" Total money=£";m/
00
 155 IF m<0 THEN GO TO 2000
 160 PRINT AT 20,0;" How many c
ps to be made? "
 170 INPUT c
 175 IF c>m/p THEN PRINT AT 12,0
;" You haven't enough money"
 176 IF c>m/p THEN GO TO 170
 180 PRINT AT 12,0;" No. of cups
made is ";c;"       "
 190 PRINT AT 20,0;" Sale price
for each cup? "
 200 INPUT l
 205 IF l>40 THEN GO TO 200
 210 PRINT AT 14,0;" Sale price=
";l;"p. per cup"
 215 PAUSE 50
 220 PRINT AT 20,0;" press a key
to open your stall"
 230 IF INKEY$="" THEN GO TO 230
 240 GO SUB 1000
 250 LET t=9
 260 PRINT AT 2,1;"TIME: ";t;":0
0 "
 265 PAUSE 30
 270 LET t=t+1
 280 IF t=13 THEN LET t=1
 290 IF t<>8 THEN GO TO 260
 300 PRINT AT 9,10; PAPER 0; INK
6;"CLOSED"
 310 PAUSE 50
 320 LET n=RND*CODE (a$(d,17))+R
ND*CODE (a$(d,18))
 330 LET a=INT (n-(1/(10-(l-p)))
*n)
```

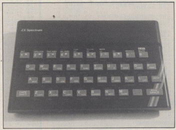
FIZZICAL

```
 340 IF a>c THEN LET a=c
 350 CLS : PRINT AT 1,2;"SALES R
EPORT"
 360 PLOT 14,159: DRAW 98,0
 370 PRINT ''" No. of cups sold="
;a
 380 PRINT ''" Money in the till
=£";(a*l)/100
 390 PRINT ''" Wholesalers bill=
£";(c*p)/100
 400 PRINT ''" RENT=£";r/100
 410 PRINT ''" PROFIT FOR THE DA
Y=£";(l*a-c*p-r)/100
 420 LET m=m+(l*a-c*p-r)
 430 IF m<0 THEN GO TO 2000
 440 LET day=day+1
 450 IF day=8 THEN GO TO 2500
 460 IF RND>.6 THEN LET p=p+1
 480 PRINT AT 20,0;" Press a key
 to continue"
 490 IF INKEY$="" THEN GO TO 490
 500 GO TO 9
 999 STOP
1000 REM Graphic set up
1001 CLS
1005 BORDER 6: PAPER 1
1010 FOR a=15 TO 21
1020 PRINT AT a,0; INK 4;"██████
██████████████████████████"
1030 NEXT a
1040 PRINT AT 7,11; PAPER 7; INK
 0;" LEMONADE "
1050 INK 6: PRINT AT 8,8;"██████
██████████"
1060 FOR a=9 TO 12
1070 PRINT AT a,9;"█";AT a,22;"█
"
1080 NEXT a
1090 FOR a=13 TO 17
1100 PRINT AT a,9;"█████████████
█"
1110 NEXT a
1115 PRINT AT 9,10;" OPEN "
1120 PRINT AT 14,10; PAPER 7; IN
K 0;"PRICE=";l;"p."
1125 INK 5
1130 PLOT 90,81: DRAW 1,-9
1140 PLOT 98,81: DRAW -1,-9
1150 PLOT 90,81: DRAW 8,0
1151 FOR a=72 TO 78
1152 PLOT 91,a: DRAW 6,0
1153 PAUSE 5
1154 NEXT a
1160 PLOT 105,81: DRAW 1,-9
1170 PLOT 113,81: DRAW -1,-9
1180 PLOT 105,81: DRAW 8,0
1181 FOR a=72 TO 78
1182 PLOT 106,a: DRAW 6,0
1183 PAUSE 5
1184 NEXT a
1185 PAUSE 30
1190 RETURN
2000 PAUSE 200: CLS
2010 PRINT AT 1,2;"FINAL REPORT"
2020 PLOT 14,159: DRAW 99,0
2030 PRINT ''" You lasted for ";
day;" days, until  you ran out o
f money"
2040 PRINT '" You didn't do very
 well,and you are not a good sho
p keeper"
2050 PRINT AT 20,0;" Press 'ENTE
R' for another go!"
2060 IF INKEY$="" THEN GO TO 206
0
2070 CLS : GO TO 5
2500 PAUSE 200
2501 CLS
2510 PRINT AT 1,2;"FINAL REPORT"
2520 PLOT 14,159: DRAW 99,0
2530 PRINT ''" You completed you
r week in      buisness and mad
e a profit of "
2535 PRINT AT 10,5;"Total money=
£";m/100
2536 PRINT AT 11,5;"minus float=
£10"
2537 PLOT 39,79: DRAW 120,0
2538 PRINT AT 13,5;"PROFIT      =
£";(m/100)-10
2550 PRINT AT 20,0;" Press 'ENTE
R' for another go!"
2560 IF INKEY$="" THEN GO TO 256
0
2570 CLS : GO TO 5
2999 STOP
3000 CLS : BORDER 1
3010 PRINT AT 1,2;"POLICE REPORT
"
3020 PLOT 14,159: DRAW 105,0
3030 PRINT ''" The police have
informed you   that vandles have
 damaged your  property. The cos
t of repairs is £20."
3040 PRINT ''" If you cannot mee
t this bill   press 'Q'. If you
want to carry on your busness pr
ess  ENTER'"
3050 IF INKEY$="q" THEN GO TO 60
00
3060 IF INKEY$="" THEN GO TO 306
0
3070 LET m=m-2000
3080 GO TO 9
5000 REM Instructions
5020 PRINT AT 1,9;"LEMONADE STAN
D"
5030 PLOT 70,159: DRAW 114,0
5040 PRINT ''" You have been pla
ced in charge   of a small lemon
ade stand in    Hyde Park,London
 for a week.   You start off wit
h £10 in your   pocket and you s
hould try to   make as much mone
y as possible."
5050 PRINT '" You are given the
report for      the day and you o
nly need to     make two decision
s:             How many cups to b
e made? and    At what price they
 should be      sold at?"
5060 PRINT ''''"     Press 'ENTER'
to open."
5070 IF INKEY$="" THEN GO TO 507
0
5080 RETURN
6000 PAUSE 200
6010 CLS
6030 PRINT AT 1,2;"BROKE"
6040 PLOT 14,159: DRAW 42,0
6050 PRINT ''" You finally ended
 up broke."
6055 PRINT ''" The police have j
ust told you     that the vandles
 are still on   the loose."
6060 PRINT AT 20,0;" Press 'ENTE
R' for another go!"
6070 IF INKEY$="" THEN GO TO 607
0
6080 GO TO 4
```

Robert Erskine

POKEING THE SPECTRUM DISPLAY

The organisation of the Spectrum display takes a bit of getting used to, but it is not as bad as it seems.



A lot of people must have felt a sudden cold chill after unpacking their brand new Spectrums and turning to page 164 of the user manual. Not only, it seems, has Mr. Sinclair decided to chop individual characters in the display file into tiny pieces and sprinkle them liberally over the screen but the screen itself has been divided into three separate blocks of eight lines each.

Before finding out whether things are really as bad as they seem I would like to expose you to the full horror of the situation. If you RUN Program 1. which POKES a byte of eight pixels into each consecutive address in the display file, you will see how the screen is built up.

Starting at the top left hand corner, the top row of each character square is filled in until the end of line seven is reached, when a jump is made to the second row of the first character in line zero, and so on. The process is then repeated with the second and third groups of eight lines until the screen is full.

SOLVING THE PROBLEM

As it turns out, the solution is not too difficult. In fact, once you have familiarised yourself with the display map, you will find that in some ways it is easier to handle than the one on the ZX81. In the first place, the display file starts at a fixed address and stays there rather than floating about in RAM. Secondly, there is no danger of POKEing characters into hidden 'newline' flags which had the interesting effect on the earlier machine of destroying your entire program when they were poked. Thirdly, since each character is composed of eight bytes, each of which is capable of holding up to eight pixels, or plot points, you can POKE any graphics you wish onto the screen without having to confine yourself to the 21 user definable graphics available in BASIC. Try running Program 1 again but this time substitute INT(RND*255) for the number 255 and you will then get some idea of the infinite possibilities.

Suppose that you wished to POKE a 2x2 block of four black squares onto the screen near the top left hand corner. Program 2 illustrates clearly in BASIC how a machine code program might be designed to do this. The first address corresponds to the first row of the first character in line 1, column 1. The variable 'byte' is the number we wish to poke into this address and is read from the data statement in line 80. 255 corresponds to the binary number 11111111, which will appear on the screen as a black bar. The 'displacement' variable, which is also read from the data statement, is the number of screen addresses which need to be added to the start address to position the first POKE, which in the first instance is zero. The second displacement is 1, which corresponds to the position immediately to the right of the start address, followed by 31 for the top row of the third character, 1 for the top of the fourth and then 223 to jump back to the second row of the first character and so on. The pause in line 60 is included to allow you to see how the whole graphic is built up.

DISPLACE HERE

It is necessary to do things in this order because every displacement value must be small enough to occupy a single byte in the data file. Although this does not matter in BASIC, it is vital in our machine code program because each item in the data table is handled one byte at a time.

Program 3 loads the machine code program which will achieve exactly the effect as Program 2 but considerably faster. RAMTOP is set at 32229 and the program is loaded from 32300 onwards. The assembled program may then be executed by entering as a direct command: RAND USR 32301.

The data statement in line 70 holds the program itself whilst line 80 holds exactly the same data as in Program 2 except that the first number is 32 rather than zero, which is the total number of bytes in the complete graphic. Number 32 is initially loaded into address 32300 which during the execution of the program keeps a running total of the number of unpoked bytes remaining.

There is a valuable bonus gained by poking characters onto the screen by this method. Because each individual byte poked can correspond to any binary number between 00000000 and 11111111 and because each character is built up byte by byte, you can build up as many user definable graphics as you wish. Try experimenting by substituting other values for the 255s in Program 3.

BLOCK CAPITALS

Finally, as a little light relief, here is a simple method of poking block colour graphics onto the screen.

The 'attributes' file in the Spectrum is happily arranged in a straightforward 32x24 format from address 22528 to 23295. By POKEing any one of these addresses you can print squares of any colour and attribute anywhere on the screen. For example, any number between 32 and 39 which is POKEd into an address will produce a green character square, whilst numbers between 96 and 103 will produce a bright green square.

Program 4 shows exactly which effects you will get as a result of POKEing every number between 0 and 255 into an attribute address. It should then be easy to adapt Program 3 to produce a machine code program which makes use of this to draw block graphics or fast changing backgrounds.

```
LD HL, 16417    First address
LD DE, 32333    Data start
LD A,(DE)       Byte count
LD (32300),A    Store
INC DE*         Next data
LD A,(DE)       Byte code
LD (HL),A       Print
LD A,(32300)
LD B,1
SUB B
RET Z           Finished?
LD (32300),A
INC DE          Next
LD A,(DE)       Displacement
LD B,0
LD C,A
ADD HL,BC       Next address
JP*             Next data
```

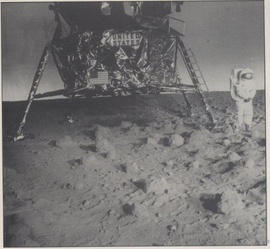
THE PROGRAMS

```
 1 REM Program 1
10 FOR x= 16384 TO 22527
20 POKE x,255
30 NEXT x

 1 REM Program 2
 2 BORDER 0
10 LET addr= 16417
20 READ displacement
30 READ byte
40 LET addr= addr + displacement
50 POKE addr,byte
60 PAUSE 25
70 GO TO 20
80 DATA 0,255,1,255,31,255,1,255,223
           255,1,255,31,255,1,255,223
           255,1,255,31,255,1,255,223
           255,1,255,31,255,1,255,223
           255,1,255,31,255,1,255,223
           255,1,255,31,255,1,255,223
           255,1,255,31,255,1,255,223
           255,1,255,31,255,1,255

 1 REM Program 3
 2 BORDER 0
10 CLEAR 32229
20 LET a =  32300
30 READ n
40 POKE a,n
50 LET a = a+1
60 GO TO 30
70 DATA 0,33,33,64,17,77,126,26,50,44,126,19,26,
        119,58,44,126,6,1,144,200,50,44,126,19,26,
        6,0,79,9,195,55,126
80 DATA 32,255,1,255,31,255,1,255,223
           255,1,255,31,255,1,255,223
           255,1,255,31,255,1,255,223
           255,1,255,31,255,1,255,223
           255,1,255,31,255,1,255,223
           255,1,255,31,255,1,255,223
           255,1,255,31,255,1,255,223
           255,1,255,31,255,1,255

 1 REM Program 4
10 LET byte = 0
20 FOR x =  22528 TO 22528+ 255
30 POKE x,byte
40 LET byte = byte+ 1
50 NEXT x
```

Mr Bish

ASTEROIDS AHEAD

Steer clear of the asteroids and keep your spaceship flying.



There you are, a lonely spaceship floating through space, when suddenly your peace is shattered by a swarm of asteroids bearing down on you at a great rate of knots. You can manoeuver your craft left using the '5' key or right with the '8' key. You also possess an energy field which you can use to deflect asteroids you cannot avoid, but beware of running out of energy — the energy field has only a limited strength and a short range.

NO STONE UNTURNED

User defined graphics have been used in this program for the asteroids, the energy screen, the spacecraft and the explosions. They are set during the initial display, thus not causing an embarrassing pause during the game.

Logic lines have been used within the program. The most well-known example of these are present in line 50, but you may like to study the less popular conditional GOTO used in line 405.

Also, the use of ATTR to

detect the presence of a white asteroid in the path of the yellow spacecraft should be noted; SCREEN $ (x,y) would not work with user defined graphics.

Line 460, which GOes to itself, is used to prevent a STOP message from spoiling the appearance of the screen. Press the Break key to stop the vicious circle.

It should also be noted that the capital letters within quotes in lines 62, 100 and 400 are user defined graphics and as such should be entered in Graphics mode.

The program is also equipped with a routine to give your score and the best score achieved, and there is also a best-ever score with the initials of the player who managed to build up that score. To operate the best-ever score option, enter:

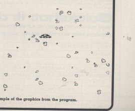
LET e= 0

as a command, and then SAVE using:

GOTO 430

The program, when LOADed, will GOTO line 1 and execute itself, complete with variables.

```
   1 BORDER 1: PAPER 1: CLS : IN
K 7: PRINT AT 0,8; FLASH 1;"ASTE
ROIDS AHEAD!": PAUSE 100: PRINT
FLASH 0;''" Steer your spacecraf
t through   the swarm of asteroi
ds"''"Press 5 for Left or 8 for
Right."''"Press 0 to activate yo
ur craft'sEnergy Screen to destr
oy          asteroids you cannot a
void"''"But BEWARE - the Screen
has onlya short range and you ha
ve only 5 energy units left"''"
Your craft will disintegrate
after 5 hits (wings=1,body=2)"''
"   HOW LONG CAN YOU SURVIVE?":
GO SUB 9000: PRINT '"          Pre
ss 8 to start": PAUSE 0: LET b=0
  10 INK 0: CLS : LET t=0: LET p
=5: LET l=15: LET l1=l: LET f=6
  50 POKE 23692,255: LET l=l+(IN
KEY$="8" AND l<29)-(INKEY$="5" A
ND l>0): IF INKEY$="0" THEN LET
f=f-1: IF f>0 THEN PRINT AT 8,l;
 INK 6;"PQR": GO TO 64
  62 IF INKEY$="0" THEN LET f=f-
1: IF f>0 THEN PRINT AT 8,l; INK
 6;"PQR"
  64 IF ATTR (8,l+2)=15 OR ATTR
(8,l)=15 THEN LET p=p-1
  66 IF ATTR (8,l+1)=15 THEN LET
 p=p-2
  67 IF p<1 THEN GO TO 400
 100 PRINT AT 7,l1;"   ";AT 21,3
1;"": PRINT : PRINT AT 7,l; INK
6;"S"; INVERSE 1;p; INVERSE 0;"T
": LET t=t+1: LET l1=l: PRINT AT
 21,RND*31; INK 7;CHR$ (RND*11+1
44);AT 21,RND*31;CHR$ (RND*11+14
4): GO TO 50
 400 FOR n=1 TO 3: PRINT AT 6,l1
;"   ";AT 7,l1; INK 2;" M ";AT 8
,l1;"   ": PAUSE 2: PRINT AT 6,l
1; INK 6;"OQU";AT 7,l1;"N N";AT
8,l1;"PQR": PAUSE 2: NEXT n: FOR
 n=6 TO 8: PRINT AT n,l1;"   ":
 NEXT n
 402 CLS : IF b<t-13 THEN LET b=
t-13
 403 INK 7: IF e<b THEN LET e=b:
 PRINT AT 6,0;"You have beaten t
he best ever   score. Enter the
HALL OF FAME"''"Press the two le
tters of your   initials": LET n
$="": FOR n=1 TO 2: PAUSE 0: LET
 n$=n$+INKEY$+".": NEXT n: CLS
 405 PRINT ;"   Score=";t-13,"Be
st Score=";b''"   Best ever Scor
e=";: PRINT FLASH 1;e;" by ";n$'
'': FOR n=1 TO 150: NEXT n: PRIN
T FLASH 0;"   Press p to play ag
ain"'''          s to save"'''
      f to finish": PAUSE 0: GO
TO (5 AND INKEY$="p")+(430 AND I
NKEY$="s")+(450 AND INKEY$="f")
 430 SAVE "asteroids" LINE 1
 450 PRINT ''''"   Thank you for
playing.  Bye"
 460 GO TO 460
9020 RESTORE 9070: FOR m=1 TO 21
9030 READ p$
9040 FOR n=0 TO 7
9050 READ a: POKE USR p$+n,a
9060 NEXT n: NEXT m: RETURN
9065 BORDER 1: PAPER 1: CLS : IN
K 7: GO TO 2
9070 DATA "a",62,33,255,129,129,
65,37,26
9080 DATA "b",112,146,136,65,123
,129,94,32
9090 DATA "c",0,0,24,36,36,56,0,
0
9100 DATA "d",0,0,48,40,48,32,0,
0
9110 DATA "e",0,24,40,40,16,0,0,
0
9120 DATA "f",112,136,136,68,40,
48,0,0
9130 DATA "g",0,0,0,56,44,16,0,0
9140 DATA "h",0,0,30,18,34,84,23
2,0
9150 DATA "i",94,177,130,228,34,
65,66,60
9160 DATA "j",0,62,65,66,34,17,1
4,0
9170 DATA "k",96,144,144,116,10,
49,65,126
9180 DATA "l",62,65,242,9,247,13
5,144,36
9190 DATA "m",145,82,16,7,244,8,
74,137
9200 DATA "n",100,24,198,0,222,0
,24,102
9210 DATA "o",0,4,34,18,1,68,50,
9
9220 DATA "p",9,50,68,1,18,34,4,
0
9230 DATA "q",0,6,170,42,73,65,1
37,137
9240 DATA "r",144,76,34,128,72,6
8,32,0
9250 DATA "s",0,3,15,15,120,200,
255,56
9260 DATA "t",0,192,240,240,30,1
9,255,28
9270 DATA "u",0,32,68,72,128,36,
76,144
```



An example of the graphics from the program.

AGF

PROGRAMMABLE JOYSTICK INTERFACE

for

Spectrum

or ZX81

ONLY 26.95 +£1.00PP

MICRODRIVE COMPATIBLE

AGF PROGRAMMABLE INTERFACE

Recognised as the only true Hardware Programmed joystick interface this product offers all the features associated with such a design.

You can use *any* Atari-compatible joystick controller with *any* software for your Sinclair Spectrum or ZX81, not just those with a joystick option.

Movement of the joystick is recognised by the computer *exactly* the same as pressing the appropriate control keys, and can therefore give the most immediate response to that movement. The hardware programmed design works with *all* possible key-reading methods, both BASIC and Machine Code.

Eight directional movement, with or without the fire button being pressed, can be achieved by only programming the left, right, up, down and fire keys required by the game.

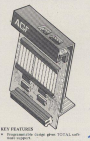
Programming is achieved by a two-digit code, which is looked up on the Programming Chart supplied, for each direction and firing button. These two numbers are then selected on a pair of leads which are clipped onto appropriately numbered strips on the interface.

Once configured this can be marked onto a Quick Reference Programming Card for storing with the game. As the programming is *not* power dependent the interface can be immediately used when next switched on.

The keyboard remains fully functional and can be used simultaneously with the joystick.

An integral rear expansion connector means there is no need to remove the interface to connect other peripherals.

NB. A recent design improvement now means that the AGF Programmable Interface works with the new Quickshot II rapid "Auto Fire" feature.



KEY FEATURES

- Programmable design gives TOTAL software support.
- Accepts Atari, Competition Pro, Wico, Starfighter, Quick Shot, Le Stick etc.
- Rear extension connector for all other add-ons.
- Free demo program and instructions.

PACKAGE CONTENTS SUPPLIED

- Programmable Interface Module as illustrated, complete with clip-on programming leads.
- Self adhesive programming chart detailing how to define which key is simulated by UP, DOWN, LEFT, RIGHT, and FIRE.

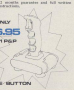
- One pack of ten Quick Reference Programming Cards for at-a-glance setting to your games requirements.

- 12 months guarantee and full written instructions.

ONLY

16.95

+£1 P&P



Quickshot II® JOYSTICK

NEW IMPROVED GRIP : BUILT-IN STABILIZING SUCTION CUPS

TRIGGER FIRE BUTTON : RAPID AUTO FIRE SWITCH : TOP FIRE BUTTON

FROM: MR/MRS/MISS

ADDRESS

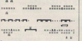
SEND C.W.O. (NO STAMP NEEDED) TO: A.G.F. HARDWARE, DEPT.**PS.**

FREEPOST, BOGNOR REGIS, WEST SUSSEX, PO22 9BR.

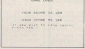
QTY	ITEM	ITEM PRICE	TOTAL
	PROGRAMMABLE INTERFACE	27.95	
	JOYSTICK(S)	17.95	
	PACK(S) QUICK REFERENCE CARDS	1.00	
	ZX81 ☐ ZX SPECTRUM ☐ Please tick *DEALER ENQUIRIES WELCOME* *EXPORT PRICES ON APPLICATION*	FINAL TOTAL	

LEARNER DRIVER

You'll have to watch out for more than the cat's eyes in this game.



A screen illustration.



```
GAME OVER

YOUR SCORE IS 100

HIGH SCORE IS 100
If you wish to play again,
press key ?
```

All you have to do in this program is to drive your car from the top left of the screen to the bottom right without hitting anything. Sound's easy enough...

When you have got the program RUNning, you find yourself looking at a page of information explaining the rules of the game. Movement of your vehicle, which is placed at the top left-hand side of the screen, is controlled by the 'l' key to move up, the 'z' key to move down, the 'm' key to move left and the 'p' key to move right. Then, you have a minute period to manoeuver your car to the bottom left-hand corner gaining as many points as you can.

MIND THAT BOLLARD

To gain points en route, you can demolish walls and fences, although you mustn't crash into a flashing bollard. Should you run over a pedestrian, or collide with one of the moving cars or trucks, you will also be told that you have crashed, given your score and asked if you would like to have another game.

You are permitted to run over cats, and as a special bonus you are awarded 100 points if you manage to run over the cat in the bottom right of the screen.

So, if you fancy an exciting drive, or you simply don't like cats, then this is the program for you.

```
   1 INK 0: PAPER 6
   2 LET hsc=0
   5 DATA 138,250,170,250,34,250
,254,80
  10 DATA 60,60,60,24,255,60,36,
102
  15 DATA 15,15,248,248,255,255,
40,56
  20 DATA 126,126,66,195,255,255
,165,231
  25 DATA 255,255,0,0,255,255,0,
0
  30 DATA 255,255,7,7,255,255,20
,28
  40 LET v=PEEK 23675+256*PEEK 2
3676
  50 FOR i=0 TO 7
  52 READ j
  54 POKE U+16+i,j
  56 NEXT i
  60 FOR i=0 TO 7
  62 READ j
  64 POKE U+40+i,j
  66 NEXT i
  70 FOR i=0 TO 7
  72 READ j
  74 POKE U+120+i,j
  76 NEXT i
  80 FOR i=0 TO 7
  82 READ j
  84 POKE U+128+i,j
  86 NEXT i
  90 FOR i=0 TO 7
  92 READ j
  94 POKE U+136+i,j
  96 NEXT i
 100 FOR i=0 TO 7
 102 READ j
 104 POKE U+144+i,j
 106 NEXT i
 120 PRINT AT 1,8;"LEARNER DRIVE
R";AT 3,2;"Drive from top left t
o bottom  right,demolishing all
walls and fences. Avoid collidin
g with     moving vehicles, pedes
```

```
trians,   and flashing bollards.
 although running over cats is p
ermitted. The final cat scores a
 bonus   100 points."
 121 OUT 254,INT (RND*32): PAUSE
 300
 122 PRINT AT 12,6;"Time Limit:
2 min."
 124 PRINT AT 14,2;"Controls  UP
  '1'   LEFT 'M'"
 126 PRINT AT 16,11;"DOWN 'Z'  R
IGHT 'P'"
 128 PRINT AT 19,7;"Press 'P' to
 play"
 130 IF INKEY$="p" THEN GO TO 15
0
 135 GO TO 120
 150 CLS
 160 DEF FN t()=INT ((65536*PEEK
 23574+256*PEEK 23673+PEEK 23672
)/50)
 165 LET t1=FN t()
 170 LET s=0
 177 LET x=0
 178 LET y=0
 189 PRINT INK 2;AT 17,5;"F";AT
17,22;"FF";AT 20,28;"F";AT 3,15;
"F";AT 8,13;"FF";AT 8,26;"F";AT
18,31;"F": REM GRAPHICS f
 190 PRINT AT 3,0;" █";AT 3,5;"*
█*█*█*█*█";AT 3,16;"█*█*█*█";AT
3,25;"█*█*█*█"
 191 POKE 22783,149: POKE 22778,
149: POKE 22773,149: POKE 22768,
149: POKE 22763,149: POKE 22753,
149
 192 PRINT AT 2,12;"#######";AT
7,12;"####";AT 7,22;"####";AT 7,
27;"####";AT 8,0;" ######";AT 8,
10;"###";AT 8,15;"####";AT 8,21;"
#####";AT 8,27;"#####"
 193 PRINT AT 13,0;" ▛▜*▛▜*▛▜*▛▜
*";AT 13,16;"▛▜▛▜▛▜***▛▜▛▜▛▜**"
 194 PRINT AT 17,0;" ####";AT 17
,6;"###";AT 17,11;"▛▀▀▀▀▜";AT 17,
19;"##*";AT 17,24;"*##*##*#";AT
18,2;"####";AT 18,21;"####";AT 1
8,26;"####"
 195 PRINT INK 1;AT 2,6;"C";AT 1
6,13;"C";AT 1,30;"C": PRINT INK
0;AT 20,30;"C": REM GRAPHICS c
 196 POKE 23105,148: POKE 23110,
148: POKE 23124,148: POKE 23129,
148: POKE 23134,148
 205 FOR n=0 TO 21
 208 IF n<=15 THEN PRINT INK 2;A
T 5,2*n+1;"Q": REM GRAPHICS q
 210 IF n<=14 THEN PRINT INK 4;A
T 5,2*n+3;"Q": IF n<=16 THEN PRI
NT AT 5,2*n-2;"  ": PRINT AT 5,
29;"  "
 215 IF n<=17 THEN PRINT INK 1;A
T 4,2*n-4;"Q": PRINT AT 4,2*n-6;
"  ": IF n>16 THEN PRINT AT 4,29
;"  "
 220 IF n<=15 THEN PRINT INK 2;A
T 10,2*n+1;"Q": PRINT AT 10,2*n-
1;"  "
 225 IF n>1 AND n<=16 THEN POKE
22880+2*n-2,163: PRINT AT 11,2*n
-4;"  ": PRINT AT 11,29;"  "
 230 IF n<=14 THEN PRINT INK 3;A
T 15,28-2*n;"PRS": IF n<=15 THEN
 PRINT AT 15,31-2*n;"   ": REM G
RAPHICS prs
 240 IF n<=10 THEN PRINT INK 1;A
T 19,3*n+1;"Q": PRINT AT 19,3*n-
2;"  "
 250 IF n<=19 THEN PRINT INK 3;A
T n+2,n+2;"Q": PRINT AT n+1,n+1;
"  "
 255 IF n<=21 THEN PRINT INK 1;A
T n,26-n;"Q": PRINT AT n-1,27-n;
"  "
 259 PRINT AT 15,0;"    "
 260 PRINT AT 21,0;"
 "
 261 PRINT AT 0,1;"
                   ": PRINT AT 20,0
;" ": PRINT AT 10,31;" ": PRINT
AT 5,31;" ": PRINT AT 19,31;" ":
 PRINT AT 11,0;" ": PRINT AT 15,
0;"  ": PRINT AT 6,0;"  "
 280 LET a=x: LET b=y
 285 IF INKEY$="z" AND x<21 THEN
 LET x=x+1
 286 IF INKEY$="1" AND x>1 THEN
LET x=x-1
 290 IF INKEY$="a" AND y>1 THEN
LET y=y-1
 291 IF INKEY$="p" AND y<31 THEN
 LET y=y+1
 300 PRINT AT a,b;" "
 310 PRINT INK 5;AT x,y;"Q"
 312 IF SCREEN$ (x,y-1)="*" THEN
 LET s=s+5
 314 IF SCREEN$ (x,y+1)="*" THEN
 LET s=s+12
 316 IF SCREEN$ (x,y-1)="#" THEN
 LET s=s+10
 318 IF SCREEN$ (x,y+1)="#" THEN
 LET s=s+12
 320 IF x=18 AND y=1 OR x=18 AND
 y=6 OR x=18 AND y=20 OR x=18 AN
D y=25 OR x=18 AND y=30 THEN GO
TO 400
 322 IF x=7 AND y=1 OR x=7 AND y
=11 OR x=7 AND y=16 OR x=7 AND y
=21 OR x=7 AND y=26 OR x=7 AND y
=31 THEN GO TO 400
 324 IF x=3 AND y=15 OR x=8 AND
y=13 OR x=8 AND y=14 OR x=8 AND
y=26 THEN GO TO 400
 326 IF x=17 AND y=5 OR x=17 AND
 y=22 OR x=17 AND y=23 OR x=20 A
ND y=26 THEN GO TO 400
 328 IF x=16 AND y=13 THEN LET s
=s+15
 330 IF x=1 AND y=30 THEN LET s=
s+15
 332 IF x=2 AND y=6 THEN LET s=s
+15
 335 IF x=4 AND y=2*n-4 OR x=4 A
ND y=2*n-3 THEN GO TO 400
 340 IF x=5 AND y=2*n+3 OR x=5 A
ND y=2*n+2 THEN GO TO 400
 344 IF x=10 AND y=2*n+1 OR x=10
 AND y=2*n+2 THEN GO TO 400
 346 IF x=15 AND y=29-2*n OR x=1
5 AND y=30-2*n THEN GO TO 400
 347 IF x=19 AND y=3*n+1 OR x=19
 AND y=3*n OR x=19 AND y=3*n-1 T
HEN GO TO 400
 348 IF x=n+2 AND y=n+2 THEN GO
TO 400
 349 IF x=n AND y=26-n THEN GO T
O 400
 350 IF x=20 AND y=30 THEN LET s
=s+100
 352 IF x=20 AND y=30 THEN GO TO
 380
 360 LET t=FN t()
 363 PRINT INK 0; PAPER 7;AT 0,2
9;t-t1;" "
 365 IF t=t1+120 THEN GO TO 380
 366 NEXT n
 370 GO TO 205
 380 CLS
 383 IF s>hsc THEN LET hsc=s
 385 PRINT FLASH 1; INK 0; PAPER
 7;AT 5,11;"GAME OVER": PRINT AT
 15,3;"If you wish to play again
,     press key c"
 387 PRINT AT 10,7;"YOUR SCORE I
S "; FLASH 1; INK 7; PAPER 0;s;"
 "
 388 PRINT AT 13,7;"HIGH SCORE I
S "; FLASH 1; INK 0; PAPER 7;hsc
;" "
 390 FOR e=1 TO 25 STEP 4
 391 BORDER INT (e/4)
 392 BEEP .5*(SIN (e*PI/2)),25
 393 BEEP .5*(COS (e*PI/2)),0
 396 IF INKEY$="c" THEN GO TO 15
0
 397 NEXT e
 399 GO TO 385
 400 CLS
 405 IF s>hsc THEN LET hsc=s
 415 PRINT INK 0; PAPER 7; FLASH
 1;AT 2,10;"COLLISION!"
 418 PRINT AT 10,7;"YOUR SCORE I
S "; FLASH 1; INK 7; PAPER 0;s;"
 "
 419 PRINT AT 13,7;"HIGH SCORE I
S "; FLASH 1; INK 0; PAPER 7;hsc
;" "
 420 PRINT AT 19,3;"If you wish
to play again,     press key c"
 421 FOR d=1 TO 29 STEP 4
 422 BORDER INT (d/4)
 423 BEEP .5,2*(SIN (d*PI/2))
 424 BEEP .5,-2*(SIN (d*PI/2))
 430 IF INKEY$="c" THEN GO TO 15
0
 431 NEXT d
 435 GO TO 415
```

ZX PROGRAMMERS . . . look no further than OCP™

Serious MACHINE CODE and BASIC programmers need the very best programming tools. Now you can buy the professional's choice.

FULL SCREEN EDITOR/ASSEMBLER (16/48K) voted THE MOST POWERFUL MACHINE CODE PROGRAMMING TOOL YET SEEN by HOME COMPUTER WEEKLY

- Editing facilities comparable to the most sophisticated word processor with MOVE, COPY and/or DELETE lines or blocks of code.
- LOCATE, CHANGE or DELETE strings/characters, full Z80 set supported, syntax check, powerful expression evaluator etc. and **"SNAKE"** a fully notated source code demonstration program. **(PLUS 80 version for the KEMPSTON CENTRONICS 80 COLUMN PRINTER INTERFACE now available).**

MACHINE CODE TEST TOOL (16/48K) tutor and de-bug program, co-resides with the FULL SCREEN EDITOR/ASSEMBLER in 48K to give a **COMPLETE MACHINE CODE DEVELOPMENT ENVIRONMENT that is second to none.**

- Allows easy entry and testing of machine coded instructions.
- Pages and displays memory registers so you actually see what's happening, displays Main and Alternate registers.

MASTER TOOLKIT (16/48K). YOUR BASIC WILL NEVER BE THE SAME AGAIN!
This program adds a whole range of really powerful commands:—

- Real time clock and alarm with off/on/set and print commands.
- BLOCK MOVE, COPY, DELETE and MERGE two lines, FIND and CHANGE character string, RENUMBER, 10 programmable keys, TRACE with execution display, VARIABLE display and dump, COMPRESS, REMKILL and PACK to minimise program bytes, CHANGE CASE upper to lower and back.

Available from selected branches of
W H SMITH, BOOTS and MENZIES and other good software stockists.

OCP™ **Software . . . simply the best**

If supply is a problem send a cheque or postal order for £9.95/program (£19.95 for the PLUS 80 version of FULL SCREEN EDITOR/ASSEMBLER)

Oxford Computer Publishing Ltd.
4 High Street, Chalfont St. Peter, Bucks. SL9 9QB

BEGINNERS

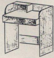
If you would like to write your own games, but find the Spectrum Manual a little heavy going, this tape is for you. It explains, using practical demonstrations, some of the more useful functions of your Spectrum, and how you can get the most from it. Tape also includes useful utilities, and a demonstration of computer graphic art. All of this plus 4 games free!! — including Fruit Machine.

Please state if your 16K or 48K Spectrum.

PRICE £4.50

MRS GRIFFIN
82 KINGSDRIVE, EGREMONT, CUMBRIA.

Home Computer "CONSOLE" in teakwood finish



Simple self-assembly
to house your:

computer, VDU/TV,
cassette player, tapes,
printer, magazines

SAFELY.

TO ORDER: Send **£49.00** plus £6.00 p&p
(delivery 21 days) to:

FORDBURY PRODUCTS
PO Box 20, Hertford SG14 2LW

CERAN SOFTWARE

Professional Software for the ZX MACHINE CODE Programmer

"CODESCAN" MACHINE CODE MONITOR

- Memory and contents display. Single step and rapid scroll up/down facility.
- Address-change facility for rapid access to any part of memory.
- Edit with automatic scroll feature.
- Run and post-run register display.
- Resides above RAMTOP leaving memory free for normal BASIC use.
- On-screen instructions and all functions shown on a single display.
- Supplied with user guide incorporating test schedule.
- Available for ZX81 16K and SPECTRUM 16/48K.
- Price £5.25 including VAT and P&P. Despatched within 7 days of order.

Please include your name and address.

Cheques/PO's payable to:

CERAN SOFTWARE
9 Parliament Road, Thame, Oxon OX9 3TE

CHESS — THE TURK from OCP™

VOTED "THE BEST CHESS OFFERING" by WHAT MICRO?

The original Turk was an eighteenth century automaton, a life-size mechanical figure resplendent in Turkish costume and seated behind a wooden cabinet on which a chess board and pieces were placed. **OCP** now offer you the twentieth century equivalent of that Turk — a chess playing computer program.

CHESS—THE TURK represents the very best in 48K ZX SPECTRUM chess programs, standard options are:—

- 6 LEVELS OF DIFFICULTY
- TWO PLAYER MODE
- RECOMMENDED MOVE
- STORING TO TAPE
- PRINT MOVE AND/OR BOARD

CHESS—THE TURK offers a whole range of additional options that have made it the definitive ZX Spectrum chess program. These enable you to:—

- CHANGE LEVELS **DURING** PLAY
- REMOVE PIECES TO EVEN THINGS UP (AS IF YOU WOULD!)
- CHANGE SIDES DURING PLAY
- PLAY AGAINST THE CLOCK
- SET UP GAMES FOR NEWSPAPER PROBLEMS

A menu driven chess program with a total of **14** options and superb graphics this program has to be seen to be appreciated.

DARE YOU PLAY THE TURK?
DON'T HESITATE . . . GET ONE TODAY

Available from selected branches of
W H SMITH, BOOTS and MENZIES and other good software stockists.

OCP™ **Software . . . simply the best**

If you experience any difficulty obtaining your copy of **CHESS—THE TURK** send a cheque or postal order for £8.95 to:

Oxford Computer Publishing Ltd.
4 High Street, Chalfont St. Peter, Bucks. SL9 9QB

David Cross

DEFENDING YOUR SPECTRUM

Can you defend the earth from the invading aliens?

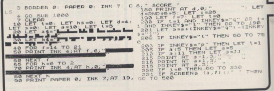
Not an easy game this! You have three ships, and your mission is to rid the planet's surface of invading aliens.

Each time you fire a missile at an invader and destroy it you will be awarded 60 points but watch out, your missiles may not always destroy the aliens. If an invader hits you then you will lose one of your ships. The only way to get more ships is to increase your score; you will get an extra ship when your score increases to 5,000, 10,000, 15,000 and 20,000.

The control keys for the game are as follows:

The '1' key moves your ship down
The 'Q' key moves your ship up
The 'P' key fires a missile
The 'T' key stops the game

Once the program has been typed in and RUN, you will have to wait four seconds for the game to start – not long really, when you consider the fate of Earth is in your hands!

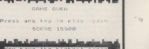


```
  3 BORDER 0: PAPER 0: INK 7: C
LS
  4 GO SUB 1000
  7 CLEAR
 10 LET t=0: LET hs=0: LET d=4:
LET s=0: LET a=10: LET l=3
 20 LET a$="
"
 30 LET b$="
"
 40 FOR f=14 TO 21
 50 PRINT INK 4;AT f,0;"
"
 60 NEXT f
 65 FOR h=0 TO 2
 70 PRINT INK 4;AT h,0;"
"
 80 NEXT h
 90 PRINT PAPER 0; INK 7;AT 19,
```

```
6;" SCORE "
 150 PRINT AT d,0;"        ": LET
d=RND*6+5: LET i=28
 190 LET f=7: LET t=0
 200 IF t=1 AND INKEY$="q" OR t=
1 AND INKEY$="1" THEN GO TO 190
 201 LET a=a+(INKEY$="q")-(INKEY
$="1")
 202 IF INKEY$="t" THEN GO TO 75
0
 203 IF INKEY$="p" THEN LET t=1
 205 IF a<5 THEN LET a=5
 210 IF a>11 THEN LET a=11
 215 PRINT AT a-1,3;"  "
 216 PRINT AT a+1,3;"  "
 218 PRINT AT a,3;" "
 220 IF t=0 THEN GO TO 230
 221 IF SCREEN$ (a,f)<>" " THEN
GO TO 500
```

```
 222 IF SCREEN$ (a,f+1)<>" " THE
N GO TO 500
 223 PRINT AT a,f;CHR$ 146
 225 BEEP .02,0
 227 PRINT AT a,f;" "
 229 LET f=f+1
 230 IF SCREEN$ (a,7)<>" " THEN
GO TO 600
 234 PRINT INK 3;AT a,5;CHR$ 144
;CHR$ 145
 235 PRINT INK 6;AT a,3;CHR$ 149
;CHR$ 148
 240 PRINT PAPER 7;AT 10,10;SCRE
EN$ (a,6)
 250 PRINT PAPER 0; INK 7;AT 19,
13;s;" "
 260 PRINT INK 4,AT 13,0;a$
 265 PRINT INK 4;AT 3,0;b$
 270 LET a$=a$(1 TO 32)+a$(1)
 275 LET b$=b$(1 TO 32)+b$(1)
 280 LET a$=a$(2 TO )
 285 LET b$=b$(2 TO )
 290 PRINT INK 7;AT d,i;CHR$ 147
;" "
 295 IF s>5000 THEN LET l=l+1
 296 IF s>10000 THEN LET l=l+2
 297 IF s>15000 THEN LET l=l+1
 298 IF s>20000 THEN LET l=l+1
 300 IF i>=13 THEN LET i=i-(RND*
1+1)
 305 IF i<13 THEN LET i=i-1
 310 IF i<=3 THEN GO TO 180
 315 IF f=31 THEN GO TO 180
 320 GO TO 200
 500 FOR h=1 TO 3
 505 PRINT INK 6;AT a,3;CHR$ 149
;CHR$ 148
 510 INK 7: PRINT AT a-1,f;CHR$
152;CHR$ 153: PRINT AT a,f;CHR$
154;CHR$ 155: PAUSE 10
 520 INK 6: PRINT AT a-1,f;CHR$
156;CHR$ 157: PRINT AT a,f;CHR$
158;CHR$ 159: PAUSE 10
 521 INK 7
 525 BEEP .1,-30: BEEP .1,-25
 530 NEXT h
 537 LET s=s+50
 540 PRINT AT a-1,f;"  ": PRINT
AT a,f;"  "
 550 GO TO 180
 600 FOR g=1 TO 3: INK 5
 602 PRINT AT a-1,4;" ";CHR$ 152
;"■";CHR$ 153
 605 PRINT AT a,4;" ";CHR$ 154;"
■";CHR$ 155: PAUSE 15
 607 INK 6
 608 BEEP .05,-30: BEEP .07,-25:
 BEEP .1,-20
 610 PRINT AT a-1,4;" ";CHR$ 156
;"■";CHR$ 157
 616 PRINT AT a,4;" ";CHR$ 158;"
■";CHR$ 159: PAUSE 10
 620 NEXT g
 627 PRINT AT a-1,4;"    "
 630 PRINT AT a,4;"    "
 632 PRINT AT a+1,4;"    ": PAU
SE 15
 635 BEEP .1,-10: BEEP .07,-4: B
EEP .2,0
 640 LET l=l-1
 650 IF l=0 THEN GO TO 750
 700 GO TO 180
 752 PRINT AT 5,10;"GAME OVER"
 755 FOR f=1 TO 3
 760 BEEP .1,-3: BEEP .1,-15: BE
EP .1,-4: BEEP .1,-5: BEEP .2,5:
 BEEP .1,-20
 770 NEXT f
 775 BEEP .1,-10: BEEP .07,-4: B
EEP .2,0: BEEP .2,2: BEEP .4,-20
 777 IF s>hs THEN LET hs=s
 780 PRINT PAPER 0; INK 7;AT 1,4
;"HIGH SCORE ";hs;" ": PRINT AT
5,10;"GAME OVER": PRINT ;AT 8,2;
"Press any key to play again": P
RINT AT 10,10;"SCORE ";s
 785 PRINT PAPER 4;AT 19,6;"
 "
 786 IF s>=hs THEN PRINT PAPER 0
; INK 7;AT 15,3;"You have the hi
ghest score"
 790 PAUSE 40
 800 PRINT AT 5,10;"         "
 810 PAUSE 30
 820 IF INKEY$<>"" THEN GO TO 7
 830 GO TO 780
1000 FOR n=144 TO 159
1010 FOR f=0 TO 7
1020 READ x: POKE USR CHR$ (n)+f
,x
1030 NEXT f
1040 NEXT n
1050 DATA 0,0,BIN 01111100,BIN 1
1111110,BIN 11111111,BIN 1111111
1,BIN 11111111,BIN 01111100
1060 DATA 0,0,0,0,0,BIN 11111111
,BIN 11111100,0
1070 DATA 0,0,0,0,0,BIN 00111110
,BIN 00111110,0
1080 DATA BIN 00111000,BIN 01010
100,BIN 01111100,BIN 00111000,BI
N 01010100,BIN 10010010,BIN 1001
0010,BIN 10010010
1100 DATA 0,0,0,BIN 00011111,BIN
 11111111,BIN 11111111,BIN 00011
111,0
1110 DATA 0,0,0,0,BIN 00000001,B
IN 00011111,0,0
1120 DATA 0,BIN 11111100,255,255
,255,255,BIN 11111110,BIN 111100
00
1130 DATA 0,0,BIN 10000000,BIN 1
1111000,BIN 11111111,BIN 1110000
0,0,0
1140 DATA 0,BIN 00001111,BIN 000
11111,BIN 00011111,BIN 00111111,
BIN 01111111,BIN 01111111,BIN 01
111111
1150 DATA 0,BIN 10000000,BIN 111
00000,BIN 11111000,BIN 11111100,
BIN 11111100,BIN 11111110,BIN 11
111110
1160 DATA BIN 01111111,BIN 01111
111,BIN 01111111,BIN 00111111,BI
N 00011111,BIN 00001111,BIN 0000
0111,0,255
1170 DATA BIN 11111110,BIN 11111
110,BIN 11111110,BIN 11111100,BI
N 11111000,BIN 11110000,BIN 1110
0000
1180 DATA BIN 00111111,BIN 00111
111,BIN 01111111,255,255,255,255
,255
1190 DATA BIN 11110000,BIN 11111
000,BIN 11111110,255,255,255,255
,255
1200 DATA 255,255,255,255,255,25
5,BIN 00111111,BIN 00011111,255
1210 DATA 255,255,255,255,255,25
5,BIN 11111110,BIN 11111100
1230 RETURN
```

Sample screen display.



```
HIGH SCORE 15900

GAME OVER

Press any key to play again

SCORE 15900

You have the highest score
```

Simon Goodwin

SPECTRAMON

Find out how your Spectrum works with this excellent program.

If you've ever wondered how your ZX Spectrum works, Spectramon (the Spectrum monitor) will make it easy for you to find out. This program will print or display the contents of ROM or RAM in numeric, character or assembly language form. Addresses may be entered in decimal or hexadecimal, and the user may select the base used for output.

Spectramon will run on a 48K Spectrum with or without a printer. The disassembler option has been written with the failings of ZX80 and ZX81 programs in mind — unlike other published listings it will handle all 694 standard Z80 instruction codes, using the standard mnemonic names and formats devised by Zilog, the firm which designed the Z80 processor used in the Spectrum.

Z80 INSTRUCTIONS

The Z80 instruction set is the most complicated of any 8-bit microcomputer. The Z80 processor was designed by a group of people who left Intel, the firm which makes the 8080 processor, to set up Zilog. The Z80 will execute any of the instructions of an 8080, plus a large number of extra ones 'tacked on' by Zilog. This approach meant that programs written to run on an 8080 would also run on a Z80 without changes. New programs could then be written using the added facilities of the Z80. That was how many early Z80 programs were produced. The BASIC interpreter used on the TRS-80 for instance, is substantially an 8080 program even though the TRS-80 has a Z80 processor Only the display and keyboard routines contain Z80 instructions since they were the last to be written.

Sinclair BASIC is written using the full features of the Z80 processor. Zilog added instructions to handle fast moving and searching of tables in memory, extra registers (internal storage) and instructions to increase the number of things that could be done with the original 8080 registers. They wanted to more than double the number of possible instructions, but there was a problem — Intel had decided to use a single byte (8 bits) to store the instruction-numbers for the 8080, and most of the 256 possible numbers were already in use.

Zilog got around this by giving four instruction numbers special meanings — instructions with one of those numbers would carry out a certain 'class' of operation, and the next byte would explain the operation required in detail. In theory, that gave Zilog plenty of possible numbers — 252 (using the remaining one-byte values) plus 1,024 (4* 256) if they were to use all of the possible two-byte instructions. In practice, they only used 694 of the 1,276 possibilities, but that's still a very large number of instructions for an 8-bit computer!

If you consult Appendix A of a Spectrum manual, you will see the standard Z80 mnemonics listed. The prefix byte 203 is used to generate add-on instructions for 'bitwise' operations — instructions which manipulate or test binary digits. The prefix byte, 221, indicates that the next instruction is an 8080 one, which would use register pair HL but must now use register IX instead. Likewise, the prefix, 253, indicates that IY should replace HL in the next instruction. If HL was in brackets in the old 8080 instruction (as in LD a, (HL)) then the Z80 version allows an offset to be applied to IX or IY before use — this is specified in an extra byte after the end of the 8080 instruction. Finally, the prefix 237 is used to indicate that the instruction following is one of a group of miscellaneous Z80 add-ons.

MONITORING THE SITUATION

If all this sounds very complicated you've probably realised why a monitor is a useful program — Spectramon will automatically convert sequences like 'EDH 7BH 3DH 5CH' into the mnemonic, LD SP, (23613). The EDH told Spectramon that it was a miscellaneous Z80 add-on instruction (EDH is 237 decimal). The 7BH corresponds to LD SP, (some address), and the 3DH 5CH corresponds to the value 23613. To check that, convert 3DH and 5CH to decimal then add the first result to the second (multiplied by 256). It's an awful lot simpler to let the computer puzzle that out than it is to work it out for every instruction by hand.

Of course, you may think that LD SP, (23613), is just as baffling as EDH 7BH 3DH 5CH — in which case, you'll have to learn a little about Z80 machine code before Spectramon becomes useful to you. Before you can investigate the ROM of a computer, you do need to understand the computer language in which it was written — assembler, in most cases. LD SP, (23613) is an assembler (or 'assembly language' or 'machine code') command. If you don't understand assembler, please don't throw this article away! It will take you no longer to learn assembler than it did to learn BASIC (it should be just as much fun too) and you can come back to Spectramon when you know more.

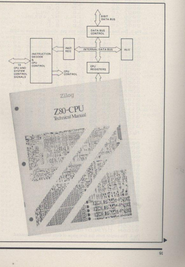
In fact, the instruction LD SP, (23613) has a very simple purpose — it tells the computer to put the number in address 23613 into the register called 'SP'. If you consult the Spectrum manual you will find out that 23613 is the 'address of item on machine stack to be used as error return', which tells you that the instruction is part of the ROM error-handler. Using a disassembled listing and the table of 'System variables' in chapter 25 of the manual, you can trace your way through the ROM, finding out what each section does.

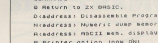
USING THE PROGRAM

Spectramon takes about 15 seconds to set itself up when first RUN. During this time, it is building a table of instruction codes for the disassembler, and once that is complete, the menu of commands will appear. Figure 1 shows this display.

To quit from the monitor, type 'Q' followed by Enter. This returns you to ZX BASIC. If you wish to disassemble a program in RAM or ROM, then you should type 'D' followed immediately by the address at which you want to start. Addresses may be entered to Spectramon in decimal or Hex — if you want to disassemble from address 126 (decimal), you could type D 126 or D007EH or D7EH — leading zeros are optional — and if you enter more than four Hex digits, only the last four will be considered. If a meaningless address is typed (such as D, DFF, D123456 or D-1) then the command will be ignored.

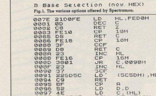
The disassembler displays the contents of memory one screenfull (21 lines) at a time. Figure 2 shows a disassembly of the first screenfull of the Spectrum ROM. The left-hand column shows the address of the instruction. It is followed on the same line by a hexadecimal representation of the instruction, and then the





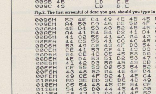
```
COMMANDS

Q Return to ZX BASIC.

D<address> Disassemble Program.

N<address> Numeric dump memory

A<address> ASCII mem. display.

P Printer option (now ON)

B Base Selection (now HEX)
```

Fig.1. The various options offered by Spectramon.



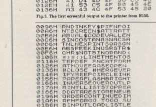
```
007E 21D0FE   LD   HL,FED0H
0081 0D       DEC  C
0082 C8       RET  Z
0083 FE10     CP   10H
0085 D8       RET  C
0086 FE18     CP   18H
0088 3F       CCF
0089 D8       RET  C
008A 23       INC  HL
008B FE16     CP   16H
008D 3801     JR   C,0090H
008F 23       INC  HL
0090 37       SCF
0091 225D5C   LD   (5C5DH),HL
0094 C9       RET
0095 BF       CP   A
0096 52       LD   D,D
0097 4E       LD   C,(HL)
0098 C4494E   CALL NZ,4E49H
009B 4B       LD   C,E
009C 45       LD   B,L
```

Fig.2. The first screenful of data you get, should you type in D126.



```
0096H   52 4E C4 49 4E 4B 45 59
009EH   A4 50 C9 46 CE 50 4F 49
00A6H   4E D4 53 43 52 45 45 4E
00AEH   A4 41 54 54 D2 41 D4 54
00B6H   41 C2 56 41 4C A4 43 4F
00BEH   44 C5 56 41 CC 4C 45 CE
00C6H   53 49 CE 43 4F D3 54 41
00CEH   CE 41 53 CE 41 43 D3 41
00D6H   54 CE 4C CE 45 58 D0 49
00DEH   4E D4 53 51 D2 53 47 CE
00E6H   41 42 D3 50 45 45 CB 49
00EEH   CE 55 53 D2 53 54 52 A4
00F6H   43 48 52 A4 4E 4F D4 42
00FEH   49 CE 4F D2 41 4E C4 3C
0106H   BD 3E BD 3C BE 4C 49 4E
010EH   C5 54 48 45 CE 54 CF 53
0116H   54 45 D0 44 45 46 20 46
011EH   CE 43 41 D4 46 4F 52 4D
0126H   41 D4 4D 4F 56 C5 45 52
012EH   41 53 C5 4F 50 45 4E 20
0136H   A3 43 4C 4F 53 45 20 A3
```

Fig.3. The first screenful output to the printer from N150.



```
0096H  RNDINKEY$PIFNPOI
00A6H  NTSCREEN$ATTRATT
00B6H  ABVAL$CODEVALLEN
00C6H  SINCOSTANASNACSA
00D6H  TNLNEXPINTSQRSGN
00E6H  ABSPEEKINUSRSTR$
00F6H  CHR$NOTBINORAND<
0106H  =>=<>LINETHENTOS
0116H  TEPDEF FNCATFORM
0126H  ATMOVEERASEOPEN
0136H  #CLOSE #MERGEVER
0146H  IFYBEEPCIRCLEINK
0156H  PAPERFLASHBRIGHT
0166H  INVERSEOVEROUTLP
0176H  RINTLLISTSTOPREA
0186H  DDATARESTORENEWB
0196H  ORDERCONTINUEDIM
01A6H  REMFORGO TOGO SU
01B6H  BINPUTLOADLISTLE
01C6H  TPAUSENEXTPOKEPR
01D6H  INTPLOTRUNSAVERA
```

Fig.4. The output from the first page of data from the command, A150.

assembly language text. After 21 lines have been displayed, the prompt 'More? (Enter = No)' will appear. Press any alphabetic or numeric key and the listing will continue on a new screen. Press the Enter key to return to the menu.

After each line is displayed, the program checks to see whether or not a key has been pressed. The Space key pauses the display, which will continue when any alphanumeric key is pressed. The Enter key causes disassembly to cease and the menu is displayed.

MAGIC NUMBER?

The third option allows display of the numeric contents of memory. Although the disassembler does this, it only lists between one and four bytes per line (depending upon the instruction). The N command allows eight bytes to be listed on each line of the display. A start address may be specified in Hex or decimal, just as for the D command.

The N command is useful for displaying the contents of tables used by a program or the ROM. Type 'N150' to see the Spectrum reserved-word table. That is where ZX BASIC stores the spellings of words such as PRINT and RETURN. The words are stored in a modified version of ASCII code — the last letter of each word has 128 (80H) added to it, to make it easy for the ROM routine which displays words to find where each one starts and ends.

If you found the numeric representation of the BASIC words rather hard to follow, you can use the command A150 to display the reserved word table in character form. The command uses 7-bit ASCII values, so that letters with 128 added to their code still print out correctly. To avoid changing colours or moving the cursor unexpectedly the, ASCII output routine displays control characters (those with a code less than 32) as full stops.

You can use the Space and Enter keys to control listings output by commands N and A, just as you would for a disassembly. Every 21 lines the

'More? (Enter = No)' message will appear before a new screen is started.

The final two commands don't output anything themselves, but they do change the output which the others generate. When you first RUN Spectramon, the message 'P' Printer option (Now ON) appears. Type the command 'P' followed by Enter and the message will become 'P Printer option (Now ON)'. If you then display memory contents (using A, N or D) the information will be sent to the printer as well as the television. Once you've finished printing, press Enter to stop the display and then use the command P to switch the printer option off again.

Notice that the printer routine does not output any lines until an entire screen-full has been generated. In fact, it deliberately avoids using the LPRINT statement to send each line to the printer. Instead it uses COPY, the ZX BASIC command which sends all of the text on the screen to the printer. That's because it is almost twice as fast to build up a full screenfull of data and then print it using COPY than it is to use LPRINT for each line as it is generated.

The printer can't stop and start very quickly and consequently the LPRINT statement is much slower than COPY — the printer must rev up and slow down 21 times (once for each line) instead of just once. In fact, the printer always outputs the last line of a group at half speed, to make sure that everything falls in the correct place when it stops. As far as it is concerned each LPRINT is the last line of a group (when there's less than 33 characters being printed).

BASE CHOICE

The final option allows the user to select the base in which numbers are output by the program. Sometimes it is useful to have numbers printed in decimal (for example, when refering to addresses mentioned in the Spectrum manual) and sometimes hexadecimal is more convenient (when displaying

Hex	Z80 Assembler	— after CB	— after ED
00	nop	rlc b	
01	ld bc,NN	rlc c	
02	ld (bc),a	rlc d	
03	inc bc	rlc e	
04	inc b	rlc h	
05	dec b	rlc l	
06	ld b,N	rlc (hl)	
07	rlca	rlc a	
08	ex af,af'	rrc b	
09	add hl,bc	rrc c	
0A	ld a,(bc)	rrc d	
0B	dec bc	rrc e	
0C	inc c	rrc h	
0D	dec c	rrc l	
0E	ld c,N	rrc (hl)	
0F	rrca	rrc a	
10	djnz DIS	rl b	
11	ld de,NN	rl c	
12	ld (de),a	rl d	
13	inc de	rl e	
14	inc d	rl h	
15	dec d	rl l	
16	ld d,N	rl (hl)	
17	rla	rl a	
18	jr DIS	rr b	
19	add hl,de	rr c	
1A	ld a,(de)	rr d	
1B	dec de	rr e	
1C	inc e	rr h	
1D	dec e	rr l	
1E	ld e,N	rr (hl)	
1F	rra	rr a	
20	jr nz,DIS	sla b	
21	ld hl,NN	sla c	
22	ld (NN),hl	sla d	
23	inc hl	sla e	
24	inc h	sla h	
25	dec h	sla l	
26	ld h,N	sla (hl)	
27	daa	sla a	
28	jr z,DIS	sra b	
29	add hl,hl	sra c	
2A	ld hl,(NN)	sra d	
2B	dec hl	sra e	
2C	inc l	sra h	
2D	dec l	sra l	
2E	ld l,N	sra (hl)	
2F	cpl	sra a	
30	jr nc,DIS		
31	ld sp,NN		
32	ld (NN),a		
33	inc sp		
34	inc (hl)		
35	dec (hl)		
36	ld (hl),N		
37	scf		
38	jr c,DIS	srl b	
39	add hl,sp	srl c	
3A	ld a,(NN)	srl d	
3B	dec sp	srl e	
3C	inc a	srl h	
3D	dec a	srl l	
3E	ld a,N	srl (hl)	
3F	ccf	srl a	
40	ld b,b	bit 0,b	in b,(c)
41	ld b,c	bit 0,c	out (c),b
42	ld b,d	bit 0,d	sbc hl,bc
43	ld b,e	bit 0,e	ld (NN),bc
44	ld b,h	bit 0,h	neg
45	ld b,l	bit 0,l	retn
46	ld b,(hl)	bit 0,(hl)	im 0
47	ld b,a	bit 0,a	ld i,a
48	ld c,b	bit 1,b	in c,(c)
49	ld c,c	bit 1,c	out (c),c
4A	ld c,d	bit 1,d	adc hl,bc
4B	ld c,e	bit 1,e	ld bc,(NN)
4C	ld c,h	bit 1,h	
4D	ld c,l	bit 1,l	reti
4E	ld c,(hl)	bit 1,(hl)	
4F	ld c,a	bit 1,a	ld r,a
50	ld d,b	bit 2,b	in d,(c)
51	ld d,c	bit 2,c	out (c),d
52	ld d,d	bit 2,d	sbc hl,de
53	ld d,e	bit 2,e	ld (NN),de
54	ld d,h	bit 2,h	
55	ld d,l	bit 2,l	
56	ld d,(hl)	bit 2,(hl)	im 1
57	ld d,a	bit 2,a	ld a,i
58	ld e,b	bit 3,b	in e,(c)
59	ld e,c	bit 3,c	out (c),e
5A	ld e,d	bit 3,d	adc hl,de
5B	ld e,e	bit 3,e	ld de,(NN)
5C	ld e,h	bit 3,h	
5D	ld e,l	bit 3,l	
5E	ld e,(hl)	bit 3,(hl)	im 2
5F	ld e,a	bit 3,a	ld a,r
60	ld h,b	bit 4,b	in h,(c)
61	ld h,c	bit 4,c	out (c),h
62	ld h,d	bit 4,d	sbc hl,hl
63	ld h,e	bit 4,e	ld (NN),hl
64	ld h,h	bit 4,h	
65	ld h,l	bit 4,l	
66	ld h,(hl)	bit 4,(hl)	
67	ld h,a	bit 4,a	rrd
68	ld l,b	bit 5,b	in l,(c)
69	ld l,c	bit 5,c	out (c),l
6A	ld l,d	bit 5,d	adc hl,hl
6B	ld l,e	bit 5,e	ld hl,(NN)
6C	ld l,h	bit 5,h	
6D	ld l,l	bit 5,l	
6E	ld l,(hl)	bit 5,(hl)	
6F	ld l,a	bit 5,a	rld
70	ld (hl),b	bit 6,b	in f,(c)
71	ld (hl),c	bit 6,c	
72	ld (hl),d	bit 6,d	sbc hl,sp
73	ld (hl),e	bit 6,e	ld (NN),sp
74	ld (hl),h	bit 6,h	
75	ld (hl),l	bit 6,l	
76	halt	bit 6,(hl)	
77	ld (hl),a	bit 6,a	
78	ld a,b	bit 7,b	in a,(c)
79	ld a,c	bit 7,c	out (c),a
7A	ld a,d	bit 7,d	adc hl,sp
7B	ld a,e	bit 7,e	ld sp,(NN)
7C	ld a,h	bit 7,h	
7D	ld a,l	bit 7,l	
7E	ld a,(hl)	bit 7,(hl)	
7F	ld a,a	bit 7,a	
80	add a,b	res 0,b	
81	add a,c	res 0,c	
82	add a,d	res 0,d	
83	add a,e	res 0,e	
84	add a,h	res 0,h	
85	add a,l	res 0,l	
86	add a,(hl)	res 0,(hl)	
87	add a,a	res 0,a	
88	adc a,b	res 1,b	
89	adc a,c	res 1,c	
8A	adc a,d	res 1,d	
8B	adc a,e	res 1,e	
8C	adc a,h	res 1,h	
8D	adc a,l	res 1,l	
8E	adc a,(hl)	res 1,(hl)	
8F	adc a,a	res 1,a	
90	sub b	res 2,b	
91	sub c	res 2,c	
92	sub d	res 2,d	
93	sub e	res 2,e	

Hex	Z80 Assembler	– after CB	– after ED
94	sub h	res 2,h	
95	sub l	res 2,l	
96	sub (hl)	res 2,(hl)	
97	sub a	res 2,a	
98	sbc a,b	res 3,b	
99	sbc a,c	res 3,c	
9A	sbc a,d	res 3,d	
9B	sbc a,e	res 3,e	
9C	sbc a,h	res 3,h	
9D	sbc a,l	res 3,l	
9E	sbc a,(hl)	res 3,(hl)	
9F	sbc a,a	res 3,a	
A0	and b	res 4,b	ldi
A1	and c	res 4,c	cpi
A2	and d	res 4,d	ini
A3	and e	res 4,e	outi
A4	and h	res 4,h	
A5	and l	res 4,l	
A6	and (hl)	res 4,(hl)	
A7	and a	res 4,a	
A8	xor b	res 5,b	ldd
A9	xor c	res 5,c	cpd
AA	xor d	res 5,d	ind
AB	xor e	res 5,e	outd
AC	xor h	res 5,h	
AD	xor l	res 5,l	
AE	xor (hl)	res 5,(hl)	
AF	xor a	res 5,a	
B0	or b	res 6,b	ldir
B1	or c	res 6,c	cpir
B2	or d	res 6,d	inir
B3	or e	res 6,e	otir
B4	or h	res 6,h	
B5	or l	res 6,l	
B6	or (hl)	res 6,(hl)	
B7	or a	res 6,a	
B8	cp b	res 7,b	lddr
B9	cp c	res 7,c	cpdr
BA	cp d	res 7,d	indr
BB	cp e	res 7,e	otdr
BC	cp h	res 7,h	
BD	cp l	res 7,l	
BE	cp (hl)	res 7,(hl)	
BF	cp a	res 7,a	
C0	ret nz	set 0,b	
C1	pop bc	set 0,c	
C2	jp nz,NN	set 0,d	
C3	jp NN	set 0,e	
C4	call nz,NN	set 0,h	
C5	push bc	set 0,l	
C6	add a,N	set 0,(hl)	
C7	rst 0	set 0,a	
C8	ret z	set 1,b	
C9	ret	set 1,c	
CA	jp z,NN	set 1,d	
CB		set 1,e	
CC	call z,NN	set 1,h	
CD	call NN	set 1,l	
CE	adc a,N	set 1,(hl)	
CF	rst 8	set 1,a	
D0	ret nc	set 2,b	
D1	pop de	set 2,c	
D2	jp nc,NN	set 2,d	
D3	out (N),a	set 2,e	
D4	call nc,NN	set 2,h	
D5	push de	set 2,l	
D6	sub N	set 2,(hl)	
D7	rst 16	set 2,a	
D8	ret c	set 3,b	
D9	exx	set 3,c	
DA	jp c,NN	set 3,d	
DB	in a,(N)	set 3,e	
DC	call c NN	set 3,h	
DD	prefixes instructions using ix	set 3,l	
DE	sbc a,N	set 3,(hl)	
DF	rst 24	set 3,a	
E0	ret po	set 4,b	
E1	pop hl	set 4,c	
E2	jp po,NN	set 4,d	
E3	ex (sp),hl	set 4,e	
E4	call po,NN	set 4,h	
E5	push hl	set 4,l	
E6	and N	set 4,(hl)	
E7	rst 32	set 4,a	
E8	ret pe	set 5,b	
E9	jp (hl)	set 5,c	
EA	jp pe,NN	set 5,d	
EB	ex de,hl	set 5,e	
EC	call pe,NN	set 5,h	
ED		set 5,l	
EE	xor N	set 5,(hl)	
EF	rst 40	set 5,a	
F0	ret p	set 6,b	
F1	pop af	set 6,c	
F2	jp p,NN	set 6,d	
F3	di	set 6,e	
F4	call p,NN	set 6,h	
F5	push af	set 6,l	
F6	or N	set 6,(hl)	
F7	rst 48	set 6,a	
F8	ret m	set 7,b	
F9	ld sp,hl	set 7,c	
FA	jp m,NN	set 7,d	
FB	ei	set 7,e	
FC	call m,NN	set 7,h	
FD	prefixes instructions using iy	set 7,l	
FE	cp N	set 7,(hl)	
FF	rst 56	set 7,a	

Fig.5. The complete Spectrum Hex codes with their corresponding assembly language mnemonics. As certain Z80 instructions are compounds starting with CBh or EDh, the two righthand columns give these. These tables have been reprinted from Appendix A of the Sinclair Spectrum manual.

address tables or working out jump offsets). Type the command 'B' to change the output base. When you first run Spectramon it will be Hex (hence the display 'B Base selection (Now=HEX)') but you can switch it to decimal with the B command. If you wish to re-select Hex output later you can 'toggle' back by typing 'B' again.

If an unknown command is entered, Spectramon will ignore it. If it is called upon to show the contents of non-existent memory (past address 65535) it will display the message 'End of Memory'. If the end of memory is encountered while the program is half-way through processing a line of numbers or ASCII characters, it will fill the rest of the line with zeros or spaces.

If you have to stop the monitor for any reason by typing Break (perhaps because your desk has melted from under the computer or the ZX printer is strangling itself) you can re-start Spectramon by entering GO TO 200 after the panic is over. So long as you've not typed LOAD, CLEAR or NEW in the meantime, the menu will appear immediately (without the 15 second wait for table set-up) and the current base (Hex or decimal) will be preserved.

MAKING A CONVERT

Spectramon is written in ZX BASIC but it should be quite easy to convert for other computers. Obviously, it will only be useful on machines which use the Z80 processor!

The Spectrum CODE function corresponds to ASC on other computers — brackets around its argument are optional in ZX BASIC. String arrays are handled rather oddly by Sinclair BASIC — the variable Z$ is set up by line 40 as having a fixed length of 32. Unused character-positions contain spaces — so that Z$ is simply used as an array of space characters by the instruction formatting routine. The array O$ contains 608 strings (numbered from one, not zero) and each string has a fixed length of nine characters (line 130).

The other string variables are normal 'Microsoft' strings — they vary in length to accommodate whatever is stored in them. ZX BASIC allows sub-strings to be extracted from a string using the 'TO' instruction — A$(1 TO 1) returns the first character of a string, corresponding to LEFT$(A$, 1) in Microsoft BASIC. If A$ is set up as 'SPECTRAMON' then A$,6,3). In short, the 'TO' instruction extracts all the characters from one position TO another, inclusive.

Spectrum BASIC allows long variable-names to be specified, and (unlike Microsoft BASIC) all the characters of a name are significant. On the Spectrum, INDEX and INDIRECT are two different, valid variables — in Microsoft BASIC they will have to be renamed, otherwise they would be treated as the same variable because they have the

Z$ —	Fixed length string of 32 spaces, used in formatting.
H$ —	Hex characters 'O'-'F' — also a local variable used in the Hex-Decimal conversion function, FN H (line 50).
LP —	'Flag' set to 1 if printout is required.
DEC —	'Flag' set to 1 if numbers must be output in decimal.
CHECK INDEX —	Line number of the routine which checks to see whether an operation could involve IX or IY.
GET INSTRUCTION —	Line number of the routine which formats a complete line of disassembler output.
MAKE TEXT —	Line number of the routine which formats a complete line of disassembler output.
BYTE VALUE —	Line number of a routine which expresses the contents of C (0-255) in C$, using the current base.
WORD VALUE —	Line number of a routine which sets up C$ with a string copy of C (0-65535) in the current base.
F$ —	String containing register names.
O$ —	String array containing the opcode text.
I,K,T —	Loop counters and temporary values.
A$ —	The command typed in by the user.
C$ —	The first character of the command.
SUB —	Line number of the chosen monitor subroutine.
LOC —	The location being examined by the monitor.
L$ —	The line of text to be output by the monitor.
I0,I1,I2 —	The instruction code and its operands.
N$ —	The name of the current index register.
S$ —	The name of the current indirect register (N$).
M$ —	The mnemonic form of the instruction.
NBYTES —	Length of instruction, in bytes.
INDEX —	Set to 1 if IX or IY are to replace HL.
INDIRECT —	Set to 1 if (IX) or (IY) are to replace (HL).
R$ —	Character within instruction mnemonic.
MODE —	Addressing mode 0-9; declares number and format of operands.
C —	Number for conversion into a decimal or Hex string.
C$ —	Number after conversion into a string.
D$ —	Part of disassembler output line.

Table 1. Variables used in Spectramon.

same first two characters. In some versions of Microsoft BASIC, neither variable name would be allowed since they both contain the key-word 'IN'. Sinclair BASIC is also unusual in that it allows spaces to occur in variable names. Table 1 shows all the variable names used in Spectramon and documents their usage.

Other systems can ignore the lines using COPY to send out a listing and simply LPRINT L$ if LP=1, printing out lines one at a time rather than *en masse*. A user defined function is set up in line 50, but it is fairly easy to code around this if your computer doesn't support that feature. FNH(H$) simply returns the decimal value of the first character in H$ — 1 for '1', 10 for 'A', 11 for 'B' and so on.

Spectramon uses a few PEEKs and POKEs which will not be required on other systems. POKE 23658,8 is a useful command which forces the Spectrum into capitals-lock (selecting a flashing 'C' as a cursor rather than a flashing 'L'). This ensures that commands are entered in capitals (unless the user purposely switches to lower-case in the course of entering a command). The location 23689 contains the number of empty lines on the Spectrum screen — when PEEK 23869 is three or less the screen is assumed to be full since the bottom two lines aren't normally used for text and a line is needed for the 'More? . . .' message.

Location 23560 contains the ASCII code of the key most recently pressed. It is set to 32 when the space bar has been pressed (or is being simulated) and 13 when Enter has been typed.

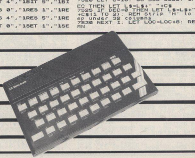
THE LAST WORD . . .

When I received my Spectrum I was convinced that I'd never get used to the keyboard. After writing, editing and typing in Spectramon I was well-practiced! Hopefully, the program also illustrates a few useful quirks of ZX BASIC, both from the BASIC and the assembler programmer's point of view. ▶

```
  40 DIM Z$(32): REM Spaces
  50 DEF FN H(H$)=CODE H$-48-7*(
H$>"@"): REM Convert hex charact
er to decimal
  60 LET LP=0: REM No printer
  70 LET DEC=0: REM Hex output
  80 LET CHECK INDEX=1800
  90 LET GET INSTRUCTION=2000
 100 LET MAKE TEXT=4000
 110 LET BYTE VALUE=5000
 120 LET WORD VALUE=5020
 130 DIM O$(608,9): REM Opcodes
 140 RESTORE
 150 READ F$
 160 LET I=1: REM Opcode No.
 170 READ O$(I)
 180 IF O$(I,1 TO 1)="1" THEN FO
R J=1 TO 7: LET O$(I+J)=O$(I): N
EXT J: LET I=I+7: REM Standard 8
-bit operand
 190 LET I=I+1: IF I<609 THEN GO
 TO 170: REM Repeat for all opco
des
 195 PRINT
 200 LET H$="0123456789ABCDEF":
REM Hex characters
 290 REM ** Command Menu
 300 CLS : PRINT "COMMANDS"
 310 PRINT '"0 Return to ZX BASI
C."
 320 PRINT '"D<address> Disassem
ble Program."
 330 PRINT '"N<address> Numeric
dump memory"
 340 PRINT '"A<address> ASCII me
m. display."
 350 PRINT '"P Printer option (n
ow ";: IF LP THEN PRINT "ON)"
 355 IF LP=0 THEN PRINT "OFF)"
 360 PRINT '"B Base Selection (n
ow ";: IF DEC THEN PRINT "DEC)"
 365 IF DEC=0 THEN PRINT "HEX)"
 390 POKE 23658,8: REM Force cap
s lock
 400 INPUT "Command?"; LINE A$
 410 IF LEN A$<1 THEN GO TO 400:
 REM Nothing typed
 415 LET C$=A$(1 TO 1): LET SUB=
0: IF C$="0" THEN CLS : STOP
 420 IF C$="B" THEN LET DEC=1-DE
C: GO TO 200: REM Reverse value
of DEC - Base flag
 425 IF C$="P" THEN LET LP=1-LP:
 GO TO 200: REM Reverse value of
 LP - Printer flag
 430 IF LEN A$<2 THEN GO TO 400:
 REM No address specified
 440 LET A$=A$(2 TO LEN A$): IF
A$(LEN A$ TO LEN A$)="H" THEN GO
 TO 2200: REM Hex address?
 450 FOR I=1 TO LEN A$: IF A$(I
TO I)<"0" OR A$(I TO I)>"9" THEN
 GO TO 400: REM Reject if addres
s is not numeric!
 460 NEXT I: LET LOC=VAL A$: REM
 Set up start address
 490 REM ** Select subroutine
 500 IF C$="D" THEN LET SUB=1000
 510 IF C$="A" THEN LET SUB=7000
 520 IF C$="N" THEN LET SUB=7500
 590 IF SUB=0 OR LOC>65535 THEN
GO TO 400: REM Ignore Silly addr
ess or command
 595 CLS : POKE 23560,0: REM Cle
ar BASIC key buffer
 600 GO SUB SUB: PRINT L$
 605 IF LOC>65535 THEN PRINT '"E
nd of memory.": POKE 23560,32: G
O TO 610: REM Pretend SPACE was
typed
 606 IF PEEK 23689<4 AND LP THEN
 COPY : REM Screen Full & print
needed
 607 IF PEEK 23689<4 AND LP=0 TH
EN PRINT "More? (Enter = NO)": P
OKE 23560,32: GO TO 610: REM Scr
een Full - pause
 610 IF PEEK 23560=32 THEN GO TO
 610: REM Wait until key<>SPACE
 612 IF PEEK 23689<3 THEN CLS :
REM Start a new screen
 615 IF PEEK 23560<>13 THEN GO T
O 600: REM Do another line unles
s ENTER was typed
 620 GO TO 200: REM Return to me
nu
1000 LET I0=PEEK LOC: REM Get op
code number
1010 IF I0=203 THEN GO TO 1200:
REM Bitwise operation (Z80)
1020 IF I0=237 THEN GO TO 1300:
REM Miscellaneous Z80 operation
1040 IF I0=221 THEN GO TO 1500:
REM Use IX instead of HL
1050 IF I0=253 THEN GO TO 1520:
REM Use IY instead of HL
1060 LET I1=PEEK (LOC+1): REM Ge
t LSB of operand
1070 LET I2=PEEK (LOC+2): REM Ge
t MSB of operand (if any!)
1080 GO SUB GET INSTRUCTION
1090 LET N$="HL": LET S$="(HL)"
1100 GO SUB CHECK INDEX: GO SUB
MAKE TEXT
1110 LET L$=L$+M$: REM Add instr
uction to line
1120 RETURN
1190 REM ** CB codes
1200 LET I0=PEEK (LOC+1)+256: RE
M Compute bitwise inst. number
1210 GO SUB GET INSTRUCTION
1220 IF M$="?" THEN GO SUB MAKE
TEXT: GO TO 1110: REM Got instru
ction
1230 LET NBYTES=2
1240 GO TO 1090: REM Finished
1290 REM ** ED codes
1300 LET I0=PEEK (LOC+1): REM Ge
t the real inst. number
1310 IF I0<64 OR (I0>127 AND I0<
160) OR I0>191 THEN LET I0=191:
REM Not a valid ED code
1315 IF I0<128 THEN LET I0=I0+32
: REM Work out position in O$
1320 LET I0=I0+417
1330 LET I1=PEEK (LOC+2): LET I2
=PEEK (LOC+3): REM Find new oper
and
1340 GO SUB GET INSTRUCTION
1350 IF M$="?" THEN GO SUB MAKE
TEXT: GO TO 1110
1360 LET NBYTES=NBYTES+1: REM Lo
nger than expected
1370 GO TO 1090
1490 REM ** DD codes
1500 LET N$="IX": REM Use IX ins
tead of HL
1510 GO TO 1530: REM Continue as
 for IY
1520 LET N$="IY": REM FD codes
1530 LET C=PEEK (LOC+2): REM Get
 offset for (IY+offset)
1540 IF C=0 THEN LET S$="("+N$+"
)"
1550 IF C>0 AND C<128 THEN GO SU
B BYTE VALUE: LET S$="("+N$+"+"+
C$+")"
1555 IF C>127 THEN LET C=256-C:
GO SUB BYTE VALUE: LET S$="("+N$
+"-"+C$+")"
1560 IF PEEK (LOC+1)=203 THEN GO
 TO 1660: REM Indexed bit operat
ion
1570 LET I0=PEEK (LOC+1): LET I1
=PEEK (LOC+2): LET I2=PEEK (LOC+
3): REM Get the equivalent HL in
st.
1580 IF I0=54 THEN LET I2=0: LET
 I1=PEEK (LOC+3): REM 8 bit indi
rect load - special case
1590 GO SUB GET INSTRUCTION
1600 LET INDEX=0: LET INDIRECT=0
: REM Not using IX or IY
1610 IF M$<>"?" THEN GO SUB CHEC
K INDEX
1620 LET NBYTES=NBYTES+INDIRECT+
INDEX: REM Adjust instruction le
ngth
1625 IF INDEX=0 THEN LET NBYTES=
NBYTES+INDIRECT: REM Extra byte
needed for offset
1630 GO SUB MAKE TEXT
1640 GO TO 1110
1660 LET I0=PEEK (LOC+3)+256: RE
```

```
M 8 bit indirect bitwise inst.
1670 GO SUB GET INSTRUCTION
1680 LET INDEX=0: LET INDIRECT=0
: REM Pretend IX & IY aren't nee
ded
1690 IF M$<>"?" THEN GO SUB CHEC
K INDEX: REM Make sure
1700 LET NBYTES=NBYTES+3*INDIREC
T
1710 GO SUB MAKE TEXT
1720 GO TO 1110
1800 LET INDIRECT=0: LET INDEX=0
: LET I=5
1810 LET I=I+1: IF I>LEN M$ THEN
 RETURN
1820 LET R$=M$(I TO I): IF R$<>"
#" AND R$<>"*" THEN GO TO 1810:
REM Check for abbreviations
1830 IF R$="*" THEN GO TO 1880
1840 LET INDEX=1: REM use IX/IY
1850 LET M$=M$(1 TO I-1)+N$+M$(I
+1 TO LEN (M$)): REM Modify inst
ruction
1860 RETURN
1880 LET INDIRECT=1: REM Use (IX
) or (IY) instead of (HL)
1890 LET M$=M$(1 TO I-1)+S$+M$(I
+1 TO LEN (M$)): REM Insert regi
ster name
1900 RETURN
2000 LET I$=U$(10+1): REM Get te
xt of opcode
2003 IF I$(LEN I$)=" " THEN LET
I$=I$(1 TO LEN I$-1): GO TO 2003
: REM Discard trailing spaces
2005 LET MODE=CODE I$-48: REM Ge
t addressing mode (if special)
2010 IF MODE<1 OR MODE>9 THEN LE
T MODE=0: GO TO 2020: REM Not sp
ecial
2015 LET I$=I$(2 TO LEN I$): REM
 Strip off the 'mode' marker
2020 FOR I=1 TO LEN I$: IF I$(I
TO I)=" " THEN GO TO 2045: REM F
ind operand field
2025 NEXT I
2030 LET A$=I$+Z$(1 TO 5-LEN I$)
: REM no operand, format neatly
2035 LET B$="": REM Operand!
2040 GO TO 2055
2045 LET A$=I$(1 TO I)+Z$(1 TO 5
-I): REM This is the opcode
2050 LET B$=I$(I+1 TO LEN I$): R
EM and this is the operand
2055 GO TO 3000+MODE*100: REM Pr
ocess in accordance with addr mo
de
2190 REM ** Convert hex to dec
2200 IF LEN A$<2 THEN GO TO 400:
 REM Must be at least 1 digit
2210 LET A$="000"+A$( TO LEN A$-
1): LET A$=A$(LEN A$-3 TO LEN A$
): REM Force 4 character field
2220 LET LOC=0: FOR I=0 TO 3: LE
T LOC=LOC+FN H(A$(LEN A$-I))*16↑
I: NEXT I: GO TO 500: REM Get va
l in LOC
2990 REM ** Use addressing mode
3000 LET NBYTES=1: REM Implied a
ddressing - no operand bytes
3010 LET M$=A$+B$: REM Build ent
ire instruction text
3020 RETURN
3100 LET NBYTES=1: REM 8 bit ope
rand in or pointed to by registe
rs
3110 IF LEN (B$)<>0 THEN LET B$=
B$+",": REM Format neatly
3115 LET K=I0-INT (I0/8)*8+1: RE
M Get operand number from inst.
3120 LET M$=A$+B$+F$(K): REM Add
 appropriate symbol
3130 IF I0=118 THEN LET M$="HALT
": REM LD (HL),(HL) doesn't exis
t - HALT takes its place
3140 RETURN
3200 LET NBYTES=2: REM 8 bit num
ber is operand
3210 IF LEN (B$)<>0 THEN LET B$=
B$+","
3220 LET C=I1: REM Get number
3230 GO SUB BYTE VALUE
3240 LET M$=A$+B$+C$: REM Build
inst.
3250 RETURN
3300 LET NBYTES=3: REM 16 bit nu
mber is operand
3310 IF LEN (B$)<>0 THEN LET B$=
B$+","
3320 LET C=256*I2+I1
3330 GO SUB WORD VALUE
3340 LET M$=A$+B$+C$
3350 RETURN
3400 LET NBYTES=2: REM Relative
jump offset is operand
3410 IF LEN (B$)<>0 THEN LET B$=
B$+","
3420 LET C=LOC-254+I1: REM Work
out absolute addr. from offset
3430 IF I1<128 THEN LET C=C+256:
 REM Can go back or forward
3440 GO TO 3330: REM Display abs
olute address of target inst.
3500 LET NBYTES=2: REM Port numb
er 0-255 is in operand byte
3510 LET C=I1
3520 GO SUB BYTE VALUE
3530 GO TO 3630
3600 LET NBYTES=3: REM Indirect
address is in operand
3610 LET C=256*I2+I1
3620 GO SUB WORD VALUE
3630 IF LEN (B$)<>0 THEN LET B$=
B$+","
3640 LET M$=A$+B$+"("+C$+")": RE
M Brackets indicate indirection
3650 RETURN
3700 LET NBYTES=2: REM IN to por
t numbered by operand
3710 LET C=I1
3720 GO SUB BYTE VALUE
3730 GO TO 3830
3800 LET NBYTES=3: REM STORE at
operand address
3810 LET C=256*I2+I1
3820 GO SUB WORD VALUE
3830 IF LEN B$<>0 THEN LET B$=",
"+B$
3840 LET M$=A$+"("+C$+")"+B$: RE
M Brackets indicate address not
value
3850 RETURN
3900 LET NBYTES=1: REM Peculiar
code
3910 LET M$="?"
3920 RETURN
3990 REM ** Build up o/p line
4000 LET C=LOC: GO SUB WORD VALU
E
4010 IF DEC THEN LET L$=C$+" ":
GO TO 4030
4020 LET L$=C$(1 TO 4)+" ": REM
Remove 'H' to shorten line
4030 LET D$="": FOR T=LOC TO LOC
+NBYTES-1
4070 LET C=PEEK T: REM Get data
4075 IF DEC THEN LET DEC=0: GO S
UB BYTE VALUE: LET DEC=1: GO TO
4090: REM Always use hex (neater
)
4080 GO SUB BYTE VALUE
4090 LET D$=D$+C$(1 TO 2): REM N
o 'H' here either
4100 NEXT T: REM Process all the
 bytes of the instruction
4110 LET L$=L$+D$+Z$(1 TO 2*(4-N
BYTES))+" ": REM Add spaces tidi
ly
4120 LET LOC=LOC+NBYTES: REM Adv
ance to next instruction
4130 RETURN
4990 REM ** Convert C to C$
5000 IF DEC THEN LET C$=STR$ C:
RETURN : REM Decimal is easy
5010 LET C$=H$(C/16+.5 TO C/16+.
5): LET C=C-INT (C/16)*16+.5: LE
T C$=C$+H$(C TO C)+"H": RETURN :
 REM 8 bit hex value
5020 IF DEC THEN LET C$=STR$ C:
RETURN
5025 LET CT=C/256: LET C$=H$(CT/
16+.5 TO CT/16+.5): LET CT=CT-IN
T (CT/16)*16+.5: LET C$=C$+H$(CT
 TO CT)
```

```
5030 LET CT=C-INT (C/256)*256: L
ET C$=C$+H$(CT/16+.5 TO CT/16+.5
): LET CT=CT-INT (CT/16)*16+.5:
LET C$=C$+H$(CT TO CT)+"H": RETU
RN : REM 16 bit hex value
5990 REM ** Z80 Instructions
6000 DATA "BCDEHL*A"
6010 DATA "NOP","3LD BC","LD (BC
),A","INC BC","INC B","DEC B","2
LD B","RLCA","EX AF,AF'","ADD #,
BC","LD A,(BC)","DEC BC","INC C"
,"DEC C","2LD C","RRCA","4DJNZ B
","3LD DE","LD (DE),A","INC DE"
6020 DATA "INC D","DEC D","2LD D
","RLA","4JR","ADD #,DE","LD A,(
DE)","DEC DE","INC E","DEC E","2
LD E","RRA","4JR NZ","3LD #","6L
D #","INC #","INC H","DEC H","2L
D H","DAA","4JR Z","ADD #,#"
6030 DATA "6LD #","DEC #","INC L
","DEC L","2LD L","CPL","4JR NC"
,"3LD SP","6LD A","INC SP","INC
*","DEC *","2LD *","SCF","4JR C"
,"ADD #,SP","6LD A","DEC SP","IN
C A","DEC A","2LD A","CCF"
6040 DATA "1LD B","1LD C","1LD D
","1LD E","1LD H","1LD L","1LD *
","1LD A","1ADD A","1ADC A","1SU
B A","1SBC A","1AND","1XOR","1OR
","1CP","RET NZ","POP BC","3JP N
Z","3JP","3CALL NZ"
6050 DATA "PUSH BC","2ADD A","RS
T 0","RET Z","RET","3JP Z","9","
3CALL Z","3CALL","2ADC A","RST 8
H","RET NC","POP DE","3JP NC","7
OUT A","3CALL NC","PUSH DE","2SU
B A"
6060 DATA "RST 10H","RET C","EXX
","3JP C","5IN A","3CALL C","9",
"2SBC A","RST 18H","RET PO","POP
 #","3JP PO","EX (SP),#","3CALL
PO","PUSH #","2AND","RST 20H"
6065 DATA "RET PE","JP (#)","3JP
 PE","EX DE,HL","3CALL PE","9","
2XOR","RST 28H","RET P","POP AF"
,"3JP PE","DI","3CALL P"
6070 DATA "PUSH AF","2OR","RST 3
0H","RET M","LD SP,#","3JP M","E
I","3CALL M","9","2CP","RST 38H"
6080 REM Z80 CB codes
6090 DATA "1RLC","1RRC","1RL","1
RR","1SLA","1SRA","9","9","9","9
","9","9","9","9","9","1SRL"
6100 DATA "1BIT 0","1BIT 1","1BI
T 2","1BIT 3"
6110 DATA "1BIT 4","1BIT 5","1BI
T 6","1BIT 7"
6120 DATA "1RES 0","1RES 1","1RE
S 2","1RES 3"
6130 DATA "1RES 4","1RES 5","1RE
S 6","1RES 7"
6140 DATA "1SET 0","1SET 1","1SE
T 2","1SET 3"
6150 DATA "1SET 4","1SET 5","1SE
T 6","1SET 7"
6160 REM Z80 ED codes (40-7F)
6170 DATA "IN B,(C)","OUT (C),B"
,"SBC HL,BC","8LD BC","NEG","RET
N","IM 0","LD I,A"
6180 DATA "IN C,(C)","OUT (C),C"
,"ADC HL,BC","6LD BC","9","RETI"
,"9","LD R,A"
6190 DATA "IN D,(C)","OUT (C),D"
,"SBC HL,DE","8LD DE","9","9","I
M 1","LD A,I"
6200 DATA "IN E,(C)","OUT (C),E"
,"ADC HL,DE","6LD DE","9","9","I
M 2","LD A,R"
6210 DATA "IN H,(C)","OUT (C),H"
,"SBC HL,HL","8LD HL","9","9","9
","RRD"
6220 DATA "IN L,(C)","OUT (C),L"
,"ADC HL,HL","6LD HL","9","9","9
","RLD"
6230 DATA "9","9","SBC HL,SP","8
LD SP","9","9","9","9","IN A,(C)
","OUT (C),A","ADC HL,SP","6LD S
P","9","9","9","9"
6240 REM More ED codes (A0-BF)
6250 DATA "LDI","CPI","INI","OUT
I","9","9","9","9"
6260 DATA "LDD","CPD","IND","OUT
D","9","9","9","9"
6270 DATA "LDIR","CPIR","INIR","
OTIR","9","9","9","9"
6280 DATA "LDDR","CPDR","INDR","
OTDR","9","9","9","9"
6990 REM ** Character dump
7000 LET C=LOC: GO SUB 5020: LET
 L$=C$+" ": FOR C=0 TO 15: IF C+
LOC>65535 THEN LET I0=32: GO TO
7030: REM Beware end of RAM
7010 LET I0=PEEK (C+LOC): IF I0>
127 THEN LET I0=I0-128: REM Stri
p 8th bit (parity or flag)
7020 IF I0<32 THEN LET I0=46: RE
M Print controls as dots
7030 LET L$=L$+CHR$ I0
7040 NEXT C: LET LOC=LOC+16: RET
URN
7490 REM ** Numeric dump
7500 LET C=LOC: GO SUB WORD VALU
E: LET L$=C$+" ": FOR I=0 TO 7:
IF LOC+I>65535 THEN LET I0=0: GO
 TO 7520: REM Beware end of RAM
7510 LET I0=PEEK (LOC+I)
7520 LET C=I0: GO SUB 5000: IF D
EC THEN LET L$=L$+" "+C$
7525 IF DEC=0 THEN LET L$=L$+" "
+C$(1 TO 2): REM Strip 'H' to ke
ep under 32 columns
7530 NEXT I: LET LOC=LOC+8: RETU
RN
```



SKULL

It's a monster...

...and it's waiting for you

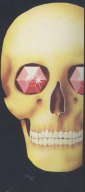
ENTER THE CATACOMBS...
HUNT THE TREASURE...
AVOID THE TRAPS...
WATCH OUT FOR SKULL!

SUPER NEW, ORIGINAL, 3-DIMENSIONAL FULL-GRAPHICS MAZE GAME FOR THE SPECTRUM 48K. HUNT THE TREASURE THROUGH MULTI-LEVEL MAZES AVOIDING HAZARDS AND DEADLY WANDERING SKULLS.

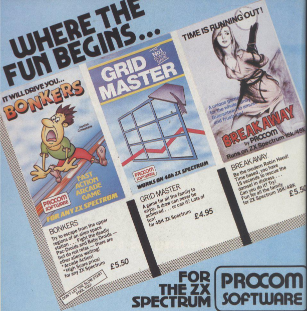
THIS GAME WILL BLOW YOUR MIND.
THERE IS NOTHING ELSE LIKE IT!

Write for details on the GRAND HIGH SCORE COMPETITION
GAMES MACHINE LTD.,
40 FRETHERNE ROAD, WELWYN GARDEN CITY, HERTFORDSHIRE.
☎ (07073) 28444/28435

GAMES MACHINE



WHERE THE FUN BEGINS...


IT WILL DRIVE YOU...
BONKERS
Joystick Compatible
FAST ACTION ARCADE GAME
PROCOM SOFTWARE
FOR ANY ZX SPECTRUM
GRID MASTER
THE No.1 BOARD GAME
PROCOM SOFTWARE
WORKS ON 48k ZX SPECTRUM
TIME IS RUNNING OUT!
A unique game for the whole family Guaranteed to amuse and frustrate!
BREAKAWAY
by PROCOM
Runs on ZX Spectrum 16k/48k

BONKERS
Try to escape from the upper regions of an alien space station . . . Fight the deadly Pac-Droids and Baby Droids — but do not relax — there are other aliens waiting!
*Arcade Action!
*High Score price!
for any ZX Spectrum
£5.50

DON'T LET THE SLOW START FOOL YOU!

GRID MASTER
A game for all the family to enjoy. A draw can never be achieved . . . or can it? Lots of fun!
for 48K ZX Spectrum
£4.95

BREAKAWAY.
Be the modern Robin Hood! Time based, you have 15 seconds to rescue the damsel in distress . . . Can you do it? Try! Fun for all the family for ZX Spectrum 16K/48K
£5.50

FOR THE ZX SPECTRUM

PROCOM SOFTWARE

- **Bonkers and Breakaway are available from Prism, Woolworths, Rumbelows, John Lewis and Debenhams.**
- **Dealer inquiries welcomed**
- **Overseas inquiries welcomed**

DO YOU HAVE A GOOD IDEA FOR A GAME? CALL US!

COMING SOON!

- **Bricky Micky**
- **Time Bug**
- **Final Frontier**
- **Bermuda Triangle**
- **Magic Orchard**

ask for details . .

HOW TO ORDER

Telephone orders
(01) 508 1216

Credit Card Holders (Access/Barclaycard only) can purchase by telephone. Please give card no., name, address and items required and your order will be despatched the same day.

BY POST

Simply write your requirements in the coupon provided, enclose your cheque P.O. or use your Access or Barclaycard. Please make cheque payable to: PROCOM SOFTWARE and post to address below. Allow 7 days for delivery.

IN PERSON

Visit our premises, at the address below for a great selection — we will be delighted to see you.

PROCOM SOFTWARE
309 HIGH ROAD · LOUGHTON · ESSEX
TEL: (01) 508 1216

POST TO: PROCOM SOFTWARE,
309 HIGH ROAD, LOUGHTON, ESSEX.

Please send me the following items:

☐ Bonkers (£5.50)
☐ Grid Master (£4.95)
☐ Breakway (£5.50 each)
☐ List of new games!

(Please tick items)

I enclose my cheque/P.O. for £ ____________

I prefer to pay with my ACCESS/BARCLAYCARD
(Delete whichever not applicable)

Card No. ____________

Signature ____________

Name ____________

Address ____________

Tel: (Day) ____________ Tel: (Eve) ____________

Credit cards valid if signed by card holder. Address above must be the same as card holder.